Universidade de Aveiro
Ano 2020

Departamento de Eletrónica,
Telecomunicações e Informática

Pedro Miguel Santos Raimundo

Gestão de um sistema fotovoltaico em instalação trifásica utilizando técnicas de Machine Learning

Management of a photovoltaic system in 3-phase installations using Machine Learning techniques

Universidade de Aveiro
Ano 2020

Departamento de Eletrónica,
Telecomunicações e Informática

Pedro Miguel Santos Raimundo

Gestão de um sistema fotovoltaico em instalação trifásica utilizando técnicas de Machine Learning

Management of a photovoltaic system in 3-phase installations using Machine Learning techniques

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica de Doutor Diogo Gomes, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente	Professor Doutor Arnaldo da Silva Rodrigues de Oliveira Professor Auxiliar, Universidade de Aveiro
vogais	Doutor Rómulo José Magalhães Martins Antão Engenheiro de Sistemas, Bosch Termotecnologia, Sa
	Professor Doutor Diogo Nuno Pereira Gomes Professor Auxiliar, Universidade de Aveiro

agradecimentos

Um agradecimento especial aos meus pais e à minha irmã, pela paciência, dedicação, apoio e encorajamento, tendo sido um suporte imprescindível, não só para a concretização deste trabalho, mas também para todo o meu percurso académico.

Agradeço, ainda, ao meu orientador Professor Doutor Diogo Gomes pelo apoio fornecido durante a realização deste projeto, bem como ao Emanuel Miranda, João Moreto e Tiago Duarte, elementos da empresa Withus, que me forneceram dados para a realização deste trabalho.

Por fim, agradeço aos meus amigos que me ajudaram e incentivaram ao longo destes anos e, também, na realização deste projeto.

palavras-chave

Instalação Trifásica, Painéis Fotovoltaicos, Machine Learning, Energia Consumida, Artificial Neural Network, Support Vector Machine, Regression Trees, Random Forest, Linear Regression, Previsão, Predição, Facebook Prophet

resumo

Foi proposta para esta dissertação, realizar uma abordagem, utilizando o Machine Learning, de forma a prever a energia consumida de cada uma das fases de uma instalação trifásica e poder alterar para a fase mais favorável, tendo em conta o seu consumo.

Para isso, as amostras de dados são retiradas diretamente dos equipamentos de medição de consumo energético. Estas amostras contêm os valores históricos das fases da instalação trifásica, com intervalos de 15 minutos.

Para prever a curto prazo, fundamental na operação dos equipamentos, foram implementados e avaliados diversos algoritmos de Machine Learning em Python 3, como, por exemplo, Artificial Neural Networks, Linear Regression, Facebook Prophet, Support Vector Machines, Random Forest e Decision Trees.

Após a previsão de cada um dos modelos, compararam-se, então, as previsões entre si com medidas de erro como RMSE, R^2 e MAPE, para determinar qual dos algoritmos é que apresenta melhor resultado.

Foi criado um outro programa em Python 3 para a escolha da melhor fase, utilizando os algoritmos que apresentaram as melhores previsões.

keywords

Three-Phase Instalation, Photovoltaic Panels, Machine Learning, Consumed Energy, Artificial Neural Network, Support Vector Machine, Regression Trees, Random Forest, Linear Regression, Forecast, Prediction, Facebook Prophet

abstract

Within the scope of this dissertation, an approach was proposed to use Machine Learning, in order to predict the consumed energy of a three-phase installation and to be able to change to the most favorable phase, taking into account its consumption.

In order to do this, data samples are taken directly from the energy consumption measuring equipment. These samples contain the historical values of the phases of the three-phase installation, with 15 minute intervals.

In order to short-term predict, crucial to operate the equipment, several Machine Learning algorithms in Python 3 were implemented and evaluated, such as, for exemple, Artificial Neural Networks, Linear Regression, Facebook Prophet, Support Vector Machines, Random Forest and Decision Trees.

After predicting each of the models, each prediction were, then, compared with error measures such as RMSE, R^2 and MAPE, to determine which algorithm has the best result.

Another Python 3 program was created to choose the best phase, using the algoritmhs that achieved the best predictions.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Siglas/Acrónimos

AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Autorregressive
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BFGS	Broyden-Fletcher–Goldfarb–Shanno
CSV	Comma-Separated Values
DL	Deep Learning
FN	False Negative
FFNN	Feed-forward Neural Network
FP	False Positive
ICT	Information and Communications Technology
k-NN	k-Nearest Neighbors
LBFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Probability Error
ML	Machine Learning
MLP	Multilayer Perceptron
MLR	Multiple Linear Regression
MSE	Mean Squared Error

nRMSE	normalised Root Mean Square Error
NWP	Netherlands Water Partnership
OOB	Out-of-Bag
RBF	Radial Based Function
ReLU	Rectified Linear Function
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SHF	Simulated Historical Forecast
SLR	Simple Linear Regression
SVC	Support Vector Classification
SVM	Support Vector Machine
SVR	Support Vector Regression
TN	True Negative
TP	True Positive
Wh	Watt-hora

1 Introdução

1.1 Contexto

A energia elétrica é um bem essencial, quer nas atividades económicas, quer nas necessidades mais simples do dia-a-dia. O mundo, cada vez mais moderno e eletrónico, faz com que as necessidades de energia elétrica sejam cada vez maiores, tornando-se num aspeto fundamental na economia.

A elevada dependência energética de recursos fósseis e as preocupações com questões ambientais representam os principais motivos que levaram ao crescimento da produção de energia através de fontes renováveis, tais como os sistemas solares, eólicos, a biomassa, entre outros.

Ao longo do tempo, tem-se vindo a assistir ao desenvolvimento da utilização da energia solar como uma fonte alternativa. Esta tem diversas aplicações, com diferentes intuitos, desde a produção de energia elétrica até aos sistemas de conforto. No que toca aos sistemas de conforto, estão englobados os sistemas de aquecimento das águas sanitárias, sistemas de aquecimento e refrigeração das habitações.

A energia elétrica, produzida em grandes estações fotovoltaicas (como, por exemplo, a estação solar fotovoltaica de Cariñena em Saragoça, a da Amareleja no Alentejo, entre outras), é injetada na rede de distribuição para consumo da população e da indústria. Podemos, ainda, considerar a produção de energia elétrica em menor escala, para o consumo próprio ou para injetar na rede, devido à localização pretendida ou a fatores económicos.

Nos transportes também é utilizada a energia solar, produzida com painéis fotovoltaicos, para movimentar os respetivos meios. Neste campo estão englobados os veículos terrestres, embarcações marítimas, aviação e no domínio aeroespacial. Contudo, é no setor aeroespacial que se dá o maior desenvolvimento na área da energia solar, sendo aplicado em satélites, torres solares e veículos espaciais.

1.2 Motivação

Com base no tema deste trabalho, torna-se importante criar um sistema que permita controlar as fases dos sistemas trifásicos, de modo a que a energia elétrica produzida pelos painéis fotovoltaicos seja aproveitada no domicílio dos consumidores ou nas empresas.

Nos sistemas elétricos de autoconsumo, qualquer energia injetada na rede elétrica não é compensada pelo comercializador, pelo que é considerada um desperdício e não se traduz em qualquer ganho para os proprietários.

Por esse motivo, os instaladores e os utilizadores deste tipo de sistema procuram maximizar o consumo da energia produzida a partir das energias renováveis, escolhendo a fase que tenha maior consumo, pois, ao fazê-lo, minimizam a perda de energia e, consequentemente, rentabilizam o investimento em sistemas de autoconsumo com poupanças de eletricidade.

As instalações elétricas trifásicas são tipicamente “desequilibradas” e isso representa um desafio para os instaladores, na medida em que o desequilíbrio entre fases não é constante e a fase ideal para injeção da energia proveniente de energias renováveis muda frequentemente.

Conjugando isto com o facto de que o perfil de produção de sistemas fotovoltaicos varia ao longo do tempo, pode-se afirmar que a decisão da fase para injeção da energia produzida não é trivial e varia ao longo do tempo, quer pela variação do consumo, quer pela variação da produção.

Foi como mestrando em Computadores e Telemática que emergiu o interesse e consequente motivação para a realização deste trabalho sobre a obtenção do melhor rendimento como consequência da alteração entre as fases da instalação trifásica.

Deste modo, uma questão se coloca, a qual identifica e define o problema a resolver neste estudo: Qual a melhor forma de alterar a fase da instalação trifásica em que se injeta a energia elétrica produzida, de modo a obter-se o melhor rendimento?

Para dar resposta a este problema, far-se-á uma explicitação de alguns conceitos relacionados com o tema deste trabalho, com base em vários autores e

investigadores neste âmbito. Assim, iremos aprofundar o modo de funcionamento de alguns algoritmos de Aprendizagem Automática, do inglês *Machine Learning* (*ML*), de forma a encontrar os que apresentem melhores resultados e, também, estudar o comportamento dos sistemas trifásicos.

Para garantir que os resultados obtidos tenham a melhor precisão, é preciso estudar diferentes métodos usados para resolver problemas de previsão e perceber as suas formas de implementação.

Neste contexto, o estudo recaiu na aplicação de vários modelos em séries temporais facultados pela empresa *Withus*[1] - Inovação e Tecnologia, Lda, sediada no concelho de Aveiro, a qual exerce atividades de consultoria em Informática.

A relevância deste estudo justifica-se pelo facto de ser possível a compreensão da importância teórica e prática que está implícita na utilização de modelos de *Machine Learning* e os seus usos práticos.

Este tema reveste-se de grande importância na comunidade científica, pois é um assunto que tem vindo a suscitar o interesse de vários autores, há já alguns anos, sobre a importância dos modelos de *Machine Learning* na previsão.

A particularidade deste estudo reside na compreensão da contribuição para a poupança de energia elétrica no domicílio, através da escolha de fases de uma instalação trifásica, dando, assim, resposta às necessidades dos consumidores na escolha automática entre as fases da instalação trifásica, obtendo o melhor rendimento.

1.3 Objetivos

Tendo em conta a questão inicial, tenciona fazer-se a descrição de um sistema que consiga alterar entre fases de uma instalação trifásica, de forma a obter-se o melhor rendimento. Este rendimento é a relação existente entre o valor da energia fotovoltaica consumida relativamente à energia produzida, a fim de obter um retorno do investimento no menor tempo possível.

Partindo da questão principal, referida no subcapítulo anterior, formularam-se as seguintes questões específicas:

1. Que modelos de *Machine Learning* adotar, de modo a encontrar os que apresentam a melhor previsão?
2. Que amostras de dados utilizar, de modo a observar diversas previsões para cada caso?
3. Que sistemas de previsão propor que permitam realizar previsões com base nos modelos usados, bem como a seleção de fases de uma instalação trifásica, tendo em conta os valores de energia previstos para cada fase?
4. Em quais ambientes ensaiar e validar as previsões?

Relembra-se que este trabalho tem como objetivo principal o desenvolvimento de um sistema capaz de gerir o escalonamento ideal da fase de injeção e da hora em que essa ligação deve ser estabelecida. O sistema deverá ter uma ligação a um comutador de fase e disponibilizar um *backoffice* de análise e monitorização do comportamento do sistema. Deste modo, pensou-se em alguns objetivos mais específicos que foram emergindo e que se apresentam a seguir:

1. Adotar modelos de *Machine Learning*, de modo a encontrar os que apresentam a melhor previsão.
2. Utilizar diferentes amostras de dados (periódicas, contínuas, descontínuas e contínuas provenientes de descontínuas), de modo a observar diversas previsões para cada caso.
3. Criar sistemas de previsão que permitam realizar previsões com base nos modelos usados, bem como a seleção de fases de uma instalação trifásica, tendo em conta os valores de energia previstos para cada fase.
4. Ensaiar e validar as previsões nos seguintes ambientes:
    - Numa instalação elétrica real, equipada com um seletor de fases que executará as instruções produzidas pelo modelo.

- Em ambiente simulado, utilizando o perfil de consumo de instalações trifásicas e comparando-o com as estimativas produzidas pelo modelo.

Assim, iniciar-se-á este estudo com o desenvolvimento do enquadramento do problema numa perspetiva teórica de acordo com a dinâmica do *Machine Learning*, nomeadamente, os diversos modelos existentes, métodos de validação e métricas de erro.

Pretende-se que as diferentes questões sejam respondidas no capítulo das conclusões.

1.4 Estrutura da Dissertação

Esta dissertação está organizada em cinco capítulos. No capítulo introdutório explicita-se o enquadramento da dissertação, os motivos que levaram à elaboração deste trabalho, a importância da previsão e o objetivo principal, bem como os objetivos específicos a serem concretizados.

No seguimento do presente capítulo introdutório, far-se-á uma revisão bibliográfica relativa ao *Machine Learning*, explicitando alguns algoritmos e as suas vantagens e desvantagens, a forma de criar e validar os diversos modelos, as avaliações do conjunto de dados para teste, de modo a verificar a precisão dos modelos.

No capítulo três é feita uma descrição sobre os dados fornecidos e metodologia utilizada na implementação de cada um dos algoritmos usados no projeto, referindo o respetivo modo de funcionamento e configuração.

O quarto capítulo incide na apresentação e discussão de resultados dos métodos usados.

Por fim, no capítulo cinco analisam-se as principais reflexões e conclusões sobre este estudo, sendo também apontadas as limitações e desafios verificados

na realização deste trabalho. Serão igualmente propostas pistas para possíveis trabalhos futuros.

Este estudo termina com as referências bibliográficas.

2 Estado de Arte

Neste capítulo são apresentados alguns algoritmos de *Machine Learning*, nomeadamente, Máquina de Vetor de Suporte, do inglês *Support Vector Machine* (*SVM*), Rede Neuronal Artificial, do inglês *Artificial Neural Network* (*ANN*), Árvore de Decisão, do inglês *Decision Tree*, Floresta de Decisão Aleatória, do inglês *Random Forest*, Regressão Linear, do inglês *Linear Regression*, Regressão Logística, do inglês *Logistic Regression*, *k-Nearest Neighbors* (*k-NN*), Modelo Autorregressivo Integrado de Médias Móveis, do inglês *Autoregressive Integrated Moving Average* (*ARIMA*) e *Facebook Prophet*. São, também, apresentados os modos de funcionamento de cada um, tendo como base artigos de alguns autores e investigadores desta área.

2.1 Previsão e *Machine Learning*

Segundo Mitchel (1997), até meados do século passado, houve uma procura de novas fontes de conhecimento devido aos desenvolvimentos tecnológicos que decorriam nesse período. Antes, a única fonte de conhecimento era simplesmente o cérebro humano, mas, por volta desse mesmo período, a humanidade começou a utilizar o processo de aprendizagem de máquinas. Estas máquinas conseguem concretizar uma enorme variedade de tarefas, inúmeras vezes, com mais precisão e rapidez que os humanos. Estas máquinas, atualmente, usam métodos de Inteligência Artificial, do inglês *Artificial Intelligence* (*AI*), que permitem reconhecer padrões complexos e as suas tendências, através de regras previamente definidas para cada tarefa. Através deste padrão de conhecimento, tornou-se mais fácil para as máquinas realizarem previsões para determinados problemas, com base em determinadas condições.

A Inteligência Artificial descreve a capacidade das máquinas de tomarem decisões, de forma semelhante aos humanos. De acordo com Makridakis et al. (2018), a Inteligência Artificial aparece como uma ferramenta versátil para a

construção de modelos de previsão. Para que isso aconteça, é necessário que estas, tal como nós, sigam regras e instruções estabelecidas anteriormente, antes da execução das mesmas, de forma a realizar operações consoante as suas previsões.

Dentro da Inteligência Artificial, existe fundamentalmente uma sub-área que considera a investigação do cérebro humano de uma forma mais detalhada, cuja designação é *Machine Learning*. Mesmo tendo semelhanças com a Inteligência Artificial, nesta sub-área é necessário ter conhecimento do comportamento humano, ou seja, a máquina precisa de ser capaz de aprender com os resultados obtidos *a priori*, mesmo que estes estejam corretos ou não. Desta forma, consegue-se fornecer uma resposta mais objetiva na resolução de problemas, através de uma lista de instruções.

Contudo, em *Machine Learning* existem dois tipos de aprendizagem comuns, referidos por Mitchel (1997), Bishop (2006), Vieira (2017) e Furão (2018):

- aprendizagem supervisionada, que trabalha com os *data-sets* (conjunto de dados) compostos por entradas e saídas. Desta forma, o programa, depois de treinar o algoritmo com os dados fornecidos anteriormente, é capaz de tomar decisões precisas ao receber novos dados. Assim, podem ser associados dois tipos de problemas nesta aprendizagem, que são os seguintes:
    - A Classificação, em que a variável de saída corresponde a uma categoria.
    - A Regressão, na qual a variável de saída representa um valor real.
- aprendizagem não-supervisionada, na qual os problemas tratados contêm *data-sets* para os processos de treino, compostos apenas por entradas. O objetivo desta aprendizagem é encontrar grupos de exemplos semelhantes entre os dados, sendo esta aplicada nos seguintes problemas:
    - Em *Clustering*, é preciso dividir os dados em categorias e grupos, mostrando relações entre eles.

- Na Associação, é necessário descobrir regras que descrevem grandes conteúdos de dados.

Para qualquer algoritmo que seja escolhido, a série temporal pode ser dividida em dois conjuntos de dados, os quais são os de treino e de teste. Os dados de treino, tal como o próprio nome indica, são comummente usados para treinar os algoritmos durante a aprendizagem. Os dados de teste, em contrapartida, servem para avaliar as previsões efetuadas, através de diversas métricas de erro. Esta divisão, apesar de não ser fixa, considera sempre o conjunto de treino contendo um número maior de valores, com um rácio de, por exemplo, 70 / 30 ou 80 / 20, em percentagem.

2.1.1 Regressões lineares e logísticas

As Regressões Lineares e Regressões Logísticas[2], como foi referenciado por Weisberg (2005) e por Mitchel (1997), têm bastantes semelhanças: ambas têm como objetivo descobrir a função que melhor se adequa aos dados treinados. Ambas usam pesos para cada fator, de forma a influenciar o resultado do fenómeno, sendo este a soma dos fenómenos mais pequenos.

As regressões logísticas podem ser usadas em diversos problemas como os de classificação entre duas classes: a partir de um conjunto de dados de entrada, é necessário saber quais pertencem a uma classe.

A Regressão Linear, referido por Weisberg (2005), é um dos algoritmos de *Machine Learning* mais simples e usados para modelar uma relação entre uma variável independente (ou resposta escalar) e uma ou mais variáveis dependentes. Para uma variável dependente, o processo é designado de Regressão Linear Simples, do inglês *Simple Linear Regression* (*SLR*), e para mais do que uma variável dependente, o processo tem o nome de Regressão Linear Múltipla, do inglês *Multiple Linear Regression* (*MLR*).

Na Regressão Linear, as relações são modeladas, usando funções de preditor linear, cujos valores dos parâmetros desconhecidos são estimados a partir dos dados. Tais modelos são chamados de modelos lineares.

Weisberg (2005) defende que os Modelos de Regressão Linear são comumente ajustados usando o método dos mínimos quadrados (*least squares*). Este método consiste em encontrar o melhor ajuste para a amostra de dados, minimizando a soma dos quadrados das diferenças entre os valores previstos e reais, como se ilustra na Figura 2.1.

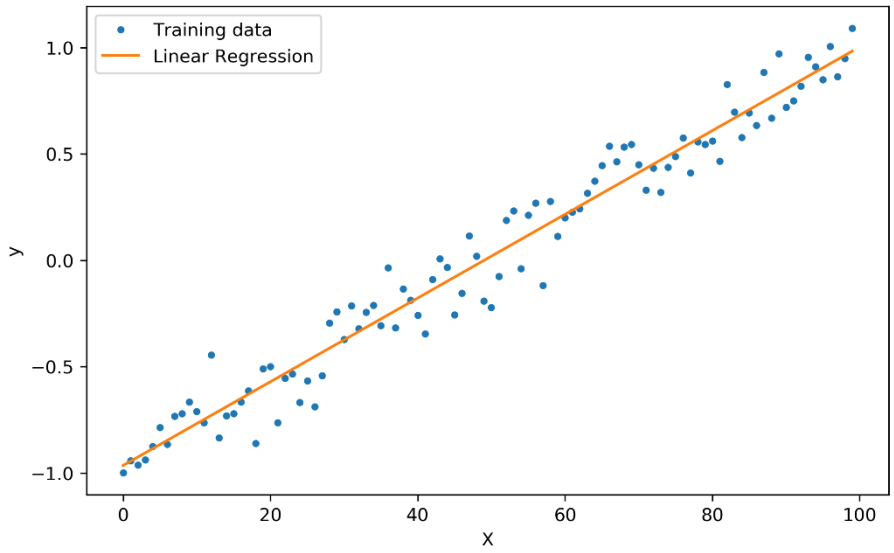

Figura 2.1 – Representação de Regressão Linear[3].

2.1.2 *Support Vector Machine*

As *SVM*, tal como são mencionadas por Tan et al. (2003), são um método de aprendizagem supervisionado com algoritmos que analisam dados usados para classificação e regressão.

Segundo Bishop (2006), referido por Antunes (2017), defende que os *SVM* são algoritmos que ultrapassam a limitação imposta pelas Regressões lineares e logísticas (o facto de a função ter uma forma linear $a_0+a_1x_1+\ldots+a_nx_n$, apesar de ser multidimensional), fazendo uma transformação não-linear na entrada.

De acordo com Antunes (2017), em Classificação do Vetor de Suporte, do ingês *Support Vector Classification* (*SVC*), torna-se possível separar duas classes com uma linha reta, denominada de hiperplano, em que anteriormente apenas se conseguia separar estas classes com uma linha não-reta. A função desta transformação chama-se *Kernel* que, através de uma entrada com n-dimensões, a transforma numa saída com mais uma dimensão (*n+1*), podendo ser possível separar as classes linearmente. Podem ser usados diversos tipos de função de *Kernel*, sendo os mais comuns a linear, polinomial e hiperbólica.

Em *SVM*, cada ponto de dados está situado num espaço *n*-dimensional (sendo *n* o número de variáveis), com o valor de cada variável pertencente a uma coordenada particular. A classificação é feita através da pesquisa do melhor hiperplano que melhor diferencia as classes. Um hiperplano é um plano de decisão que separa um conjunto de objetos em diferentes classes, conforme é apresentado na Figura 2.2.

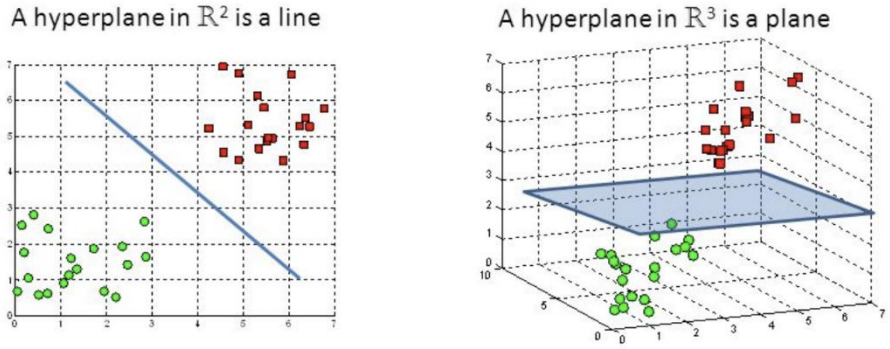

Figura 2.2 – Representação de um hiperplano no espaço em *SVM*[4].

Para problemas de Regressão, deve-se usar o algoritmo Regressão do Vetor de Suporte, do inglês *Support Vector Regression* (*SVR*). É semelhante ao *SVM* e usa uma função de *Kernel* para transformar um conjunto de dados não-linear num linear, através do cálculo de um hiperplano equivalente. Pode ser usado para resolver problemas de regressão, em vez de problemas de classificação, no qual o objetivo do mesmo é encontrar a função que melhor se adapta aos dados.

Para Tan et al. (2003), as vantagens das *SVM* são:

- Eficácia num espaço de muitas dimensões.

- Efetividade em situações em que o número de dimensões é maior que o número de casos.
- Uso de um subconjunto de pontos de treino na função de decisão (chamados vetores de suporte), para que seja mais eficiente em termos de memória.
- Versatilidade: diferentes funções de *Kernel* podem ser especificadas para cada função de decisão, tais como linear, polinomial e esférica, utilizadas também por Antunes (2017).

As desvantagens das *SVM*, referidas por Tan et al. (2003), são:

- Se o número de variáveis for maior que o número de casos, é crucial evitar excessos na escolha das funções de *Kernel*.
- Não estimam probabilidades diretamente. Estas podem ser calculadas usando uma validação, mas com o uso de um número elevado de recursos.

2.1.3 *Artificial Neural Network*

Outro tema fundamental para estudo de *Machine Learning* são as Redes Neuronais Artificiais.

Para Mitchel (1997) e Bishop (2006), o objetivo nesta área, tal como o nome indica, é criar uma rede neuronal para resolver problemas complexos de *Machine Learning*, com uma configuração e execução semelhante à dos nossos cérebros, ou seja, aprendem a executar instruções sem precisarem de ajuda exterior.

Este tipo de rede é constituído por nós (ou neurónios), interligados entre si, que enviam informação entre os mesmos, tendo em conta os valores de entrada, para resolver problemas de complexidade não-linear e executar as tarefas que chegam aos nós de saída. Todas estas unidades são semelhantes entre si, constituídas por neurónios artificiais, que reagem uns com os outros na rede e contêm ligações entre todos, chamadas de sinapses, definindo a sua interação.

Podem ter múltiplas entradas e saídas, sendo aplicados em problemas de regressão e classificação.

As ligações (sinapses), numa pequena escala (*input-synapse-perceptron-output*), ligam as entradas aos neurónios, multiplicando-os com o peso descrito em w_i^k, na qual w^k é o conjunto de peso numa camada k. O neurónio retira a informação obtida por cada sinapse, ligado ao mesmo, $x_i w_i$, e efetua a sua soma.

De modo a chegar ao resultado, é necessária uma função de ativação, podendo esta ser *ReLU* (*Rectified Linear Function*), *Sigmoid*, entre outros, para introduzir não-linearidade. O resultado então é a saída do neurónio. Sem uma função de ativação não-linear, as redes neuronais podem ser simplificadas por apenas uma única camada.

Em qualquer camada da rede neuronal, é necessário que todas as entradas estejam ligadas com todos os neurónios. No entanto, se uma entrada não for muito relevante para o problema em si, o seu peso vai ser mais reduzido.

Outra consideração relaciona o processo de treino, tendo em conta o algoritmo de aprendizagem e a rapidez do mesmo. Sendo métodos de otimização, estes são usados em muitos problemas de *Machine Learning* para encontrar o custo mínimo da função, tais como *Adam* e Descida de Gradiente Estocástico, do inglês *Stochastic Gradient Descent* (*SGD)*, descrito por Kingma & Ba (2014), o *BFGS* (*Broyden-Fletcher-Goldfarb-Shanno*), por Liu & Nocedal (1989), e *Adamax*, descrito por Kingma & Ba (2014).

Para Yona et al. (2007), o *Feed-Forward Neural Network* (*FFNN*) permite a transição de sinais da entrada para a saída. Não existe nenhum *feedback*, ou seja, a saída de cada camada não afeta nenhuma entrada nessa mesma camada. Costumam ser redes simples, que associam as entradas com as saídas, e são maioritariamente usadas no reconhecimento de padrões. São ideais para modelar relações entre as variáveis de entrada com uma ou mais variáveis de saída, isto é, são apropriadas para qualquer problema funcional onde se precisa de saber como é que um número de variáveis de entrada afeta a variável de saída.

As Redes Neuronais Recorrentes, do inglês *Recurrent Neural Network* (*RNN*), podem conter sinais a atravessar ambas as direções da mesma, introduzindo

ciclos. Estas redes são fortes e extremamente complicadas de gerir. As computações efetuadas anteriormente nas entradas são novamente colocadas na rede, o que oferece um tipo de memória. Estas redes são dinâmicas, isto é, o seu estado muda constantemente até alcançar um ponto de equilíbrio e mantêm-se neste estado até serem detetadas modificações, recomeçando este processo novamente.

Uma arquitetura da Rede Neuronal é comummente dividida em 3 partes, e está ilustrada na Figura 2.3:

- Camada de Entrada de Dados, do inglês *Input Layer*, que recebem o conjunto de dados de treino e de teste para processamento.
- Camada Escondida, do inglês *Hidden Layer*, que é constituída por uma ou mais camadas, influenciando o desempenho da rede, de modo a evitar o subdimensionamento.
- Camada de Saída dos Dados, do inglês *Output Layer*, na qual o número de nós de saída depende do problema de *Machine Learning*.

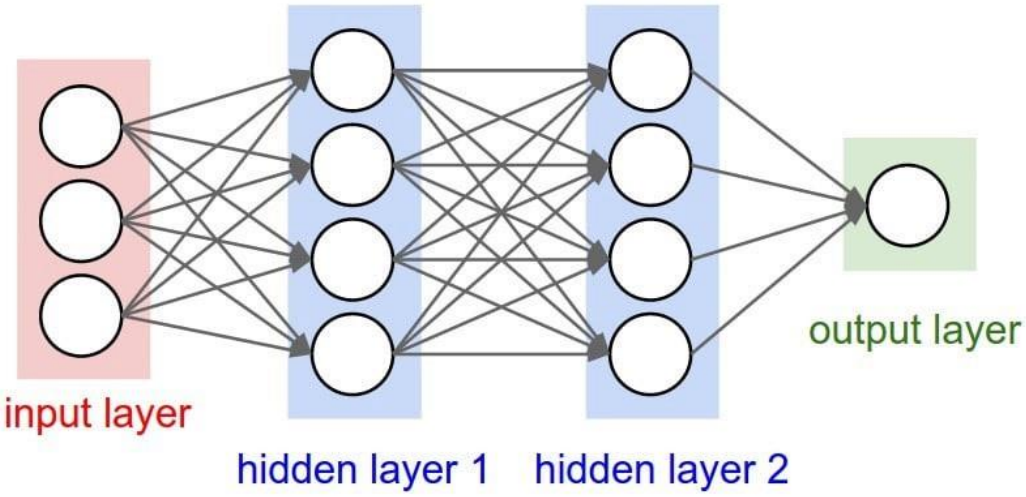

Figura 2.3 – Representação de uma Rede Neuronal Artificial[5].

2.1.4 *k-Nearest Neighbors*

Este método pode ser aplicável tanto a problemas de classificação como de regressão.

Como Mitchel (1997) refere, o algoritmo tem como objetivo guardar os exemplos de treino, ou seja, quando se avalia uma nova instância, o algoritmo realiza comparações dos seus valores com os exemplos guardados em memória, escolhendo o valor mais próximo das instâncias estudadas.

Este processo encontra-se ilustrado na Figura 2.4.

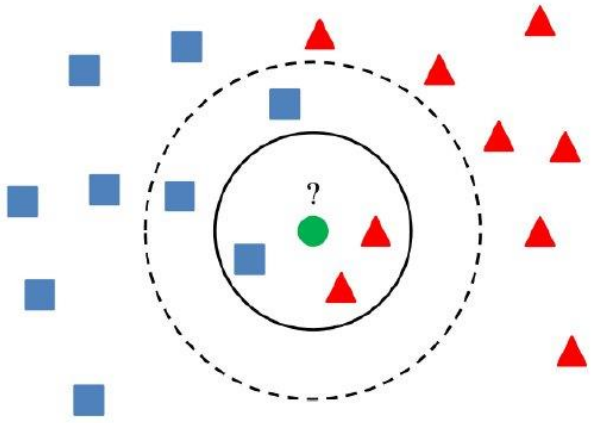

Figura 2.4 – Ilustração do algoritmo de *k-NN*, por Alaliyat (2008).

O círculo verde é a amostra a ser classificada. Os quadrados azuis e triângulos vermelhos ilustram amostras de duas classes diferentes no conjunto de treino. Se *k* for três, a classe da amostra será atribuída tendo em conta a decisão dos três vizinhos mais próximos (ilustrados pelo círculo preto). Se *k* for cinco, os cinco vizinhos mais próximos serão considerados (ilustrados pelo círculo preto a tracejado).

Contudo, este tipo de método torna-se muito lento se forem fornecidas múltiplas instâncias num curto espaço de tempo, visto que não tem uma função de alvo global. Torna-se, então, necessário avaliar cada nova instância introduzida. Por isso, este método não necessita de treino inicial sem ser por memorização, sendo fácil de implementar. Assim, a função de destino é, então, substituída por funções locais mais simples.

2.1.5 *Decision Tree*

Decision Trees, descritas por Francisco (2015), são utilizadas principalmente para definir estratégias de decisão. Este algoritmo, descrito por Oshiro (2013), coloca o *data-set* num conjunto de decisões binárias, sendo este conjunto de decisões a criação de uma árvore, que se divide em dois ramos em cada nó. Para alcançar a decisão final, é necessário percorrer todos os nós de decisão, até chegar ao nó da folha.

Descritas com detalhe por Criminisi et al. (2014) e por Tan et al. (2003), apresenta bastantes analogias com a vida real e influencia uma ampla área de *Machine Learning*, abrangendo tanto a classificação como a regressão[6].

Para considerar o espaço das variáveis, Francisco (2015) e Bishop (2006) referem a importância de começar no nó raiz da árvore e aplicar consecutivamente um critério de divisão aos restantes nós, até chegar ao nó folha. A divisão necessita de ser efetuada utilizando a variável que é mais representativa no problema, fazendo, de seguida, uso das restantes variáveis para dividir os restantes conjuntos de dados de entrada, como se ilustra na Figura 2.5.

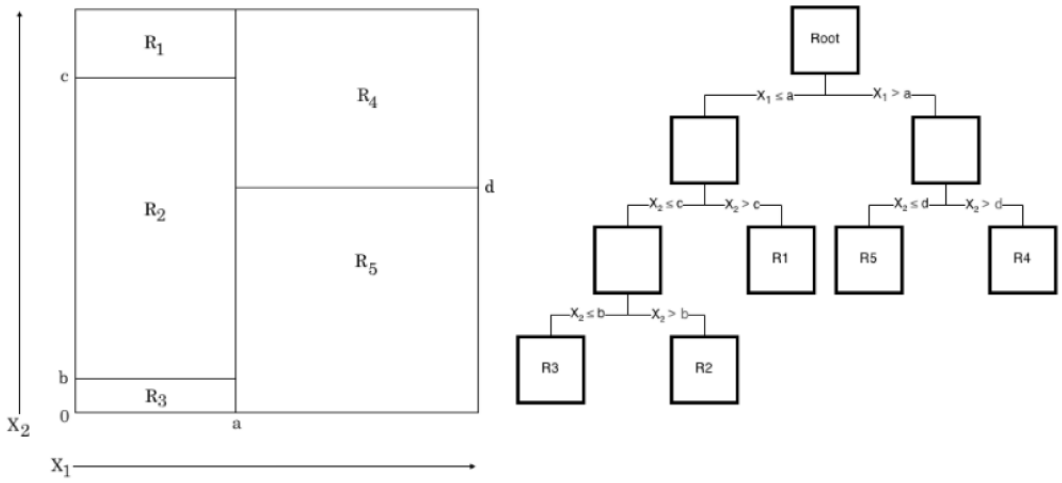

Figura 2.5 – À esquerda, apresenta-se uma divisão do conjunto de dados em cinco classes diferentes. À direita, uma árvore de decisão capaz de classificar um elemento numa das cinco classes (por Dubinets (n.d.)).

Para Criminisi et al. (2014) e Tan et al. (2003), as vantagens de usar este algoritmo são:

- Simples de entender, interpretar e visualizar.
- Executa implicitamente a seleção de recursos.
- Pode lidar com dados numéricos e categóricos, bem como lidar com problemas de vários *outputs*.
- Requerem relativamente pouco esforço do utilizador para a preparação de dados.
- Os relacionamentos não lineares entre parâmetros não afetam o desempenho da árvore.

Os mesmos autores Criminisi et al. (2014) e Tan et al. (2003) referem as seguintes desvantagens:

- Podem criar árvores muito complexas, comprometendo a generalização dos dados. Tal facto é conhecido como *overfitting*, situação que pode ocorrer quando o algoritmo modela os padrões gerais e também ruído específico do próprio conjunto de dados.
- Podem ser instáveis, porque pequenas variações nos dados podem resultar na criação de uma árvore completamente diferente. Esta situação é denominada de variância, a qual precisa de ser reduzida, usando o método de *Boosting*.
- Podem aparecer algoritmos "gananciosos", em que os dados dos mesmos não estão "treinados" de forma a garantir uma árvore de decisão com os melhores resultados. Isso pode ser atenuado pelo treino de várias árvores.

2.1.6 *Random Forest*

Para diversos problemas de classificação e regressão, uma só árvore de decisão pode não ser suficiente para realizar previsões com precisão. Esta

estrutura de árvore, explicado por Tong (2013), é sensível a pequenas alterações no conjunto de dados e, como as divisões feitas no espaço de variáveis são paralelas ao eixos, não são ótimas.

Para resolver o problema, é necessária a utilização de outro algoritmo de *Machine Learning*, o *Random Forest*. Para Francisco (2015) e Finlay (2014), este algoritmo é um conjunto de *Decision Trees*, no qual cada árvore utiliza um subconjunto distinto do conjunto de dados para treino, adicionando aleatoriedade ao processo de aprendizagem e resultando em árvores diferentes. Assim, o *output* da floresta é uma combinação dos *outputs* gerados por cada árvore. Este processo encontra-se representado na Figura 2.6.

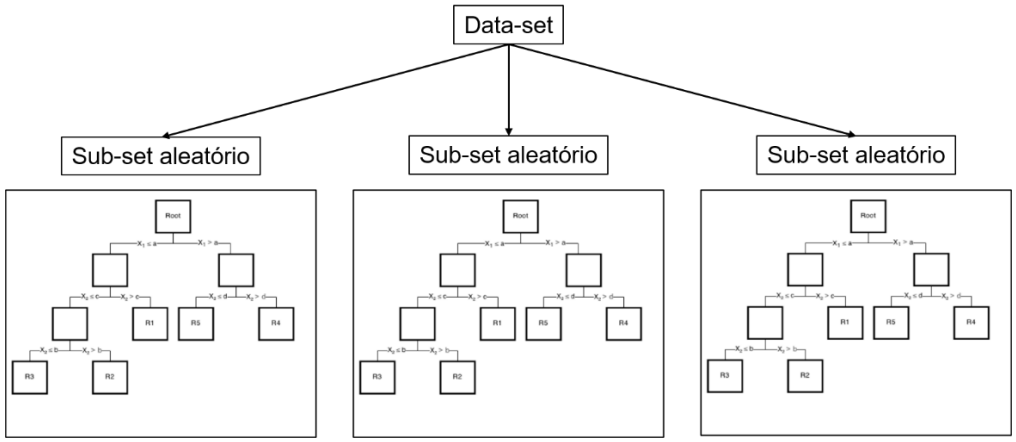

Figura 2.6 – Representação de uma Floresta de Decisão Aleatória (adaptada de Dubinets (n.d.)).

O *Random Forest* é outro algoritmo de *Machine Learning*, descrito por Criminisi et al. (2014) e Breiman (2001). É um algoritmo de aprendizagem *Ensemble* para classificação e regressão, construindo um vasto conteúdo de *Decision Trees* durante a execução.

As *Random Forests* corrigem o *overfitting* realizado pelas árvores de decisão no seu conjunto de treino. Tal como referiram Criminisi et al. (2014), Breiman (2001) e Bishop (2006), existem dois métodos *Ensemble*:

- *Bagging*, que indica que cada classificador é treinado com um exemplo do conjunto de dados em paralelo e a decisão global é feita de tal modo que a classe prevista é a que obtiver a maioria dos votos.
- *Boosting*, que utiliza também uma votação para decidir a classe, tendo em conta os pesos dos modelos de acordo com os seus desempenhos. É um procedimento iterativo em cascata, ou seja, novos classificadores são influenciados pelo desempenho dos anteriores, de forma a encontrar erros nas classificações feitas por classificadores anteriores.

O princípio básico deste algoritmo é combinar aprendizes fracos e gerar um algoritmo com bom desempenho, ou seja, cada aprendiz fraco é uma árvore de decisão treinado em paralelo com:

- Um conjunto de treino aleatório.
- Uma seleção aleatória de variáveis em cada nó.
- *Out-of-Bag* (*OOB*): um conjunto de dados para testar o desempenho.

Para a sua regra de decisão cada excerto de dados é classificado por todas as árvores e a decisão final é feita com a maioria dos votos.

Para estimar o teste de erro, é necessário que, para cada exemplo num *data-set*, seja preciso prever a classe, mas só usar as árvores que pertençam ao conjunto *OOB* que contenham esse exemplo.

Descritas por Breiman (2001) e Criminisi et al. (2014), as vantagens do *Random Forest* passam por ter um bom desempenho, baixa complexidade (classificadores simples) e erros *Out-of-Bag* (*cross-validation* não é necessária). As desvantagens do mesmo são a sensibilidade dos mesmos a ruído (maioria com *Boosting*) e são difíceis de entender num conjunto de classificadores.

2.1.7 *Facebook Prophet*

Como Taylor & Letham (2017) descrevem, os utilizadores registados no Facebook são capazes de criar eventos, convidar outros utilizadores e interagir. Ao observar os dados diários de eventos criados no *Facebook*, Taylor & Letham (2017) conseguiram afirmar a existência de uma tendência sazonal visível: ciclos semanais e anuais, com uma diminuição entre o Natal e Ano Novo.

Assim, estas amostras de dados, afirmadas por Harvey & Peters (1990), contêm três elementos importantes: tendência, sazonalidade e os feriados. Podem ser descritas com a seguinte fórmula:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{2.1}$$

onde *g(t)* é uma função que indica a tendência dos valores durante o intervalo de tempo, *s(t)* corresponde a mudanças periódicas, como, por exemplo, mudanças semanal e anual, e por último *h(t)* representa os efeitos dos feriados que potenciam irregularidades por um certo período. O termo de erro ϵ_t significa algumas mudanças que não são acomodadas pelo modelo.

Segundo Taylor & Letham (2017), o *Facebook Prophet*[7] é um algoritmo que, tal como os anteriores, permite prever dados de séries temporais com base num modelo aditivo, no qual as tendências não-lineares se ajustam à sazonalidade anual, semanal e diária, tendo, ainda, suporte para calendário de férias. As previsões são efetuadas, utilizando um horizonte, denominado *H*. Este horizonte significa o número de dias que queremos saber sobre a previsão.

Para estes autores, no entanto, não se consegue utilizar um método como o *cross-validation* neste algoritmo, porque não se pode particionar o conjunto de dados. Assim, foram criados *k* subconjuntos de dados de valores históricos simulados, do inglês *Simulated Historical Forecasts* (*SHFs*), de modo a que estes horizontes estejam contidos nos conjuntos de valores históricos e que se consiga avaliar o erro total. Este processo, descrito por Tashman (2000), usa uma

sequência de datas e não a realização de uma previsão por todos os valores históricos. Uma vantagem de ser usado é a poupança de potência computacional.

Descritos por Taylor & Letham (2017), os *SHFs* simulam os erros que seriam cometidos se os autores tivessem utilizado este método de previsão em datas específicas do passado. Contudo, é essencial estar consciente de dois problemas quando se usa esta metodologia:

- Quanto maior for o número de previsões simuladas, mais próximas são as estimativas de erro. Se tivermos uma previsão simulada de cada dia dos valores históricos, então estas previsões não são suscetíveis a mudar consideravelmente. Da mesma forma, se não obtivermos previsões suficientes, então teremos poucas observações sobre erros de previsão.
- Os métodos de previsão podem resultar melhor com muitos *data-sets* ou não. Um grande conjunto de dados de valores históricos pode resultar em piores previsões quando o modelo não está especificado e quando se encontra muito ajustado no passado.

2.1.8 *ARIMA*

ARIMA, descrito por Wan et al. (2015) e Lopes (2019), é uma metodologia generalizada de Modelos Autorregressivos de Médias Móveis, do inglês *Autoregressive Moving Average* (*ARMA*), que pertence a uma família de modelos lineares e flexíveis, usado na modelação de séries temporais sazonais e não sazonais. Pode representar diversos tipos de séries temporais como autorregressivas, do inglês *autorregressive* (*AR*), média móvel, inglês *moving average* (*MA*) ou ambas, e têm melhores resultados, quando são aplicados em casos onde os dados mostram indícios de serem não-estacionários.

Esta teoria de modelo foi desenvolvida por diversos investigadores e aplicada por Box et al. (1977). Através do processo iterativo de construção de três etapas,

sendo estas a identificação e diagnóstico do modelo e estimativa de parâmetros, a metodologia *Box-Jenkins* foi comprovada como sendo uma aproximação eficaz na modelação de séries temporais.

Este modelo consiste na identificação de ordens de diferenciação (*d, D*) e das partes autorregressiva (*p, P*) e de média móvel (*q, Q*), apresentado por:

$$ARIMA(p, d, q)(P, D, Q)_m \tag{2.2}$$

2.1.9 Síntese dos modelos

A Tabela 2.1 apresenta uma síntese com algumas das características dos algoritmos atrás referidos.

Não estão referenciadas, na tabela, algumas características para alguns modelos, pois na documentação consultada não se encontrou informação relevante para classificar os referidos algoritmos.

Tabela 2.1 – Comparação entre os diversos algoritmos de aprendizagem supervisionada de *Machine Learning*[8].

	Regressão Logística	**Regressão Linear**	***SVM***	***ANN***	***k-NN***	***Decision Tree***	***Random Forest***	***Facebook Prophet***	***ARIMA***
Tipo de problema	Classificação	Regressão	Ambos	Ambos	Ambos	Ambos	Ambos	Regressão	Regressão
Velocidade de treino	Rápido	Rápido	Rápido	Lento	Rápido	Rápido	Moderado	-	-
Velocidade de previsão	Rápido	Rápido	Rápido	Rápido	Depende do n (tamanho da amostra)	Rápido	Moderado	-	-
Executa bem com amostras pequenas?	Sim	Sim	-	Não	Não	Não	Não	Sim	Não
Separa bem o sinal do ruído?	Não	Não	Não	Sim	Não	Não	Sim (a não ser que o ruído seja elevado)	Sim	Sim
Paramétrica?	Sim	Sim	Não	Não	Não	Não	Não	-	-

2.2 Avaliação do modelo

Apesar de o principal objetivo dos algoritmos de *Machine Learning* terem semelhanças entre si, é necessário avaliar cada um dos modelos considerados.

É preciso ter em conta que não existe um só modelo que seja considerado o ideal, dado que alguns podem ser melhores que os outros a encontrar o padrão geral, ignorando picos ocasionais, e outros podem fornecer uma melhor compreensão de certos eventos.

Antunes (2017) descreve que, quando se mede o desempenho de uma experiência estatística como a previsão do consumo, existem duas grandes dimensões:

- Descrição da tendência do fenómeno;
- Comportamento do mesmo com erros e ruído possíveis.

Para avaliar o desempenho geral de um certo método, calcula-se a diferença entre cada valor da previsão $\hat{x}_{pred}$ e cada valor real x, dado pela seguinte expressão de erro:

$$e_t = x_t - \hat{x}_{pred_t} \tag{2.3}$$

Assim, ao observar cada par observação-predição associado, consegue-se calcular a média de erro, através da divisão da soma de cada erro pelo número total de erros.

Contudo, este erro médio não demonstra a precisão, visto que é possível obter uma média de erro equivalente a zero, mesmo que os erros não tenham o valor de zero. Apenas se consegue identificar, se as previsões efetuadas estiverem acima ou abaixo do alvo.

Para Theocharides et al. (2018), considerando Y_t o valor observado e P_t a previsão efetuada, ambos para o instante *t*, as métricas mais utilizadas na obtenção do erro são as seguintes:

- Erro Médio, do inglês *Mean Error* (*ME*): média das diferenças entre os valores observados e previstos.

$$ME = \frac{1}{n}\sum_{t=1}^{n} Y_t - P_t \tag{2.4}$$

- Erro Absoluto Médio, do inglês *Mean Absolute Error* (*MAE*): média das diferenças absolutas entre os valores observados e previstos.

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |Y_t - P_t| \tag{2.5}$$

- Erro Quadrático Médio, do inglês *Mean Squared Error* (*MSE*): média do quadrado das diferenças entre os valores observados e os valores previstos.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} (Y_t - P_t)^2 \tag{2.6}$$

- Erro da Raiz Quadrática Média, do inglês *Root Mean Squared Error* (*RMSE*): representação da raiz quadrada do valor de *MSE*.

$$RMSE = \sqrt{MSE} \tag{2.7}$$

- Erro da Raiz Quadrática Média normalizada, do inglês *normalised Root Mean Square Error* (*nRMSE*): normalização de *RMSE*, definindo o alcance (diferença entre o valor máximo e mínimo), para facilitar a comparação entre diferentes modelos.

$$nRMSE = \frac{RMSE}{Y_{max} - Y_{min}} \tag{2.8}$$

- Erro Percentual Absoluto Médio, do inglês *Mean Absolute Percentage Error* (*MAPE*): percentagem da média absoluta da diferença entre os valores observados e os previstos. Para previsões muito baixas, o erro percentual não pode exceder 100%, mas para previsões muito altas, não há limite superior para o erro percentual.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - P_t}{Y_t}\right| \times 100 \tag{2.9}$$

- *SS* (*Skill Score*): percentagem que descreve o grau de associação do modelo previsto, referido por $RMSE_{predicted}$, ao modelo de referência, descrito como $RMSE_{reference}$, sendo 100% uma previsão perfeita e 0% uma previsão igual ao modelo de referência.

$$SS = \left(1 - \frac{RMSE_{predicted}}{RMSE_{reference}}\right) \times 100 \tag{2.10}$$

Normalmente, um modelo que tenha um valor de *RMSE* mais baixo é considerado melhor que outros, ainda que existam outras métricas usadas para casos mais específicos, como é o caso do *ME*, *MAE* e *MAPE*.

As diferenças e a média de erro podem ser utilizadas para análise de estimativas, podendo confirmar se estas estão próximas das observações efetuadas ou não. Quanto menores forem estes desvios, melhor será a previsão e, também, esta previsão irá ajustar melhor se forem adicionados ruído e *outliers*.

Para Renaud & Victoria-Feser (2010), outra métrica a ser considerada é o Coeficiente de Determinação, do inglês *Coefficient of Determination* (R^2), que representa a dispersão em volta da linha de regressão. O resultado desta métrica é representado no intervalo $]-\infty; 1]$, no qual representará uma melhor previsão se o resultado for próximo de *um*, sendo *um* o resultado de um ajuste perfeito. É uma relação entre a diferença das observações e as previsões com a diferença entre as observações e a média. A fórmula é a seguinte:

$$R^2 = 1 - \frac{\sum_{i=0}^{n_{samples}^{-1}}(x - \hat{x})^2}{\sum_{i=0}^{n_{samples}^{-1}}(x - \bar{x})^2} \tag{2.11}$$

A fórmula é apenas a probabilidade de os novos valores serem corretamente previstos. Tem um valor máximo de um, quando não existem erros de previsão, mas esse valor diminui à medida que o erro é superior.

2.3 Validação do modelo

Para garantir que os resultados obtidos pelos algoritmos de *Machine Learning* sejam fiáveis e significativos, é necessária uma abordagem que assegure a qualidade de cada um dos modelos, com o objetivo de encontrar anomalias nos modelos ou nos conjuntos de dados.

Assim, é importante que se analise a exatidão e a confiança de um dado algoritmo. Deste modo, fazendo uso das variáveis extraídas dos dados fornecidos, consegue-se comprovar a exatidão, ou seja, se o modelo em análise está em conformidade com os resultados obtidos. Por outro lado, a confiança avalia o modo do comportamento do modelo em diferentes amostras de dados.

Para validar a capacidade de um modelo de *Machine Learning*, podem ser usados diferentes métodos, alguns dos quais se descrevem a seguir:

- *k-fold Cross-Validation*. De acordo com Martins (2011), este método serve para validar algoritmos de aprendizagem. Descreve-se como a divisão do conjunto de dados em *k* divisões igualmente distribuídas, sendo uma delas para teste e as restantes para treino. Podem ser efetuados *k* testes, para cada uma das *k* divisões. Referido por Antunes (2017), com estes resultados pode-se observar que estes modelos não sofrem de *overfitting*.
- *Leave-One-Out Cross-Validation*. Neste caso particular, descrito por Diamantidis et al. (2000) e referenciado por Ferreira (2010), o conjunto de dados é dividido de tal forma que existe só um elemento para o processo de teste. Assim, com *l* o tamanho do *data-set*, o treino é

realizado com *l-1* elementos, sendo depois efetuado o teste usando o último elemento.

- *Hold-Out Percentage Split*. Referido por Ferreira (2010) e explicado por Diamantidis et al. (2000), neste método o conjunto de teste é escolhido aleatoriamente, com cerca de 20% ou 30% dos elementos disponíveis. Os restantes elementos são usados para o treino do algoritmo e depois validados com o conjunto de teste.

Também é possível avaliar a robustez do método com dados automaticamente gerados. Com este tipo de abordagem, é possível comparar valores de previsão com os valores esperados, através do cálculo dos valores de uma função definida.

2.4 Trabalhos relacionados

O *Machine Learning* não é um tema propriamente recente na atualidade. Aliás, existem vários artigos e dissertações, de entre os quais alguns são abordados a seguir.

O autor Antunes (2017) analisa o consumo de água ao longo do tempo e efetua uma previsão para o futuro, tendo em conta os métodos e parâmetros de *Machine Learning* usados. Para obter maior precisão na previsão dos métodos de *Machine Learning*, foram também utilizados, pelo autor, vários tipos de amostras de dados, como a temperatura, com diferentes intervalos de tempo (12h, 24h e 48h), entre outros. Além disso, os diferentes algoritmos usados, pelo autor, foram:

- *ANN*, combinado com três funções de ativação (*Identity*, *Sigmoid* e *ReLU*), três números de camadas (2, 5 e 8) e três algoritmos de otimização (*SGD*, *LBFGS* e *Adam*).
- *SVR*, com três tipos de *Kernel* (radial, linear e polinomial) e dois valores de tolerância (0.01 e 0.001).

- *K-Nearest Neighbors*, que depende bastante do número de vizinhos considerados relevantes para o problema. Foram usados três números de vizinhos (2, 5 e 8) e duas funções de peso (uniforme e de distância).
- *Random Forest*, com combinações de três números de árvores (2, 8 e 15) e são analisados dois números de amostras necessárias para dividir (2 e 8).

Estes modelos, combinados com diversos tipos de *data-sets*, foram comparados com a metodologia *ARIMA*, de modo a observar os algoritmos que obtiveram melhores resultados.

Para Yona et al. (2007) o foco das previsões de potência através dos sistemas fotovoltaicos reside nas Redes Neuronais Artificiais, no qual os autores explicam com detalhe o uso entre *Feed-Forward Neural Network* e *Recurrent Neural Network* para diversos tipos de dados e as suas vantagens e desvantagens. Através do uso de métricas referidas anteriormente, os erros são minimizados ao usar a metodologia *Recurrent Neural Network*.

No artigo de Theocharides et al. (2018), é previsto o consumo futuro de energia com vários algoritmos, de modo a obter o menor erro possível. Foram utilizados vários tipos de amostras de dados como, por exemplo, de temperatura, para além das amostras de valores de energia histórica. Foram, ainda, aplicados diversos algoritmos de *Machine Learning* como *ANN*, *SVR* e *Regression Trees*. Com os resultados obtidos para cada tipo de previsão, foram comparados, entre si, através da utilização de diferentes métricas, de modo a obter-se maior precisão nos resultados. Além disso, verificou-se que o desempenho da predição dos modelos de *FFNN* alcançaram os menores valores de *MAPE* e *nRMSE*, o que significa que, para este problema, estes modelos conseguem prever a energia/potência de uma forma mais precisa do que os modelos *SVR* e *Regression Trees*.

Um outro artigo dos autores Wan et al. (2015), refere o uso de *data-sets* históricos de energia gerada, meteorologia, temperatura, humidade, velocidade do vento, entre outros, e ainda dados de *Netherlands Water Partnership* (*NWP*) de forma a obter uma combinação com os diversos tipos de previsão seguintes:

- muito curto-prazo (previsão para os cinco minutos seguintes). Útil para controlo de instalações fotovoltaicas.

- curto-prazo (previsões entre 48 e 72 horas). Pode ser utilizada na resolução de problemas que envolvem o mercado de eletricidade ou a operação dos sistemas.
- médio-prazo (previsões a uma semana). É útil na programação da manutenção de sistemas fotovoltaicos, transformadores ou linhas de transmissão.
- longo-prazo (previsões a meses ou anos). Pode ser aplicada no planeamento de instalação de painéis fotovoltaicos no domicílio.

O *Machine Learning* aplica-se ainda, para além das áreas mencionadas anteriormente, na previsão de vendas no setor do Retalho, situação descrita por Lopes (2019) e Ramos et al. (2015), em que os estudos destes autores explicam a utilidade desta ferramenta em contexto real. Segundo Lopes (2019), para realizar as melhores previsões foram propostas três abordagens diferentes para resolver o problema: *ARIMA*, *Multilayer Perceptron* (*MLP*) e uma combinação das duas anteriores.

A partir do *Machine Learning* surge uma classe mais abrangente da mesma, cujo nome é dado por Aprendizagem Profunda, do inglês *Deep Learning* (*DL*). Como Rijo (2017) refere, esta área concentra um conjunto de técnicas e ferramentas mais específicas para realizar abstrações. Quer isto dizer que o modelo pode ignorar a fase de extração das variáveis, utilizando um conjunto de dados em bruto. Este tipo de abordagem é utilizado em diversas aplicações, como o Reconhecimento de Voz, Visão Computacional e Deteção do Movimento, entre outras.

Existem outras áreas onde é relevante existir *Machine Learning*, como, por exemplo, a Classificação de Sinais de Trânsito. Os autores Silva et al. (2015) explicam a importância deste feito na área da indústria automóvel, para que o condutor seja avisado de ações impróprias e perigosas, de modo a que consiga tomar decisões com base na informação processada pelos algoritmos usados em *Machine Learning*.

O *Random Forest* é um modelo muito usado em *Machine Learning*, sendo uma alternativa para processos de classificação e regressão. Segundo Francisco (2015), o uso de *Random Forest* e *Decision Trees* é importante na deteção do

cancro, através da visualização do mesmo com sistemas específicos. Um outro trabalho relacionado com a área da Saúde é a previsão de risco de doenças como Alzheimer, referido por Araújo (2013), no qual se encontram estes modelos referidos anteriormente.

O *Random Forest* pode ser utilizado para diversos problemas de classificação, mesmo com dados desequilibrados. Como descreve Chen et al. (1999), dados desequilibrados são os que são constituídos com pelo menos uma das classes com uma menor quantidade de dados. Para estes problemas é necessário existir uma correção da classe "rara", com menor quantidade de dados. Outros algoritmos não conseguem realizar uma previsão com bons resultados quando este tipo de problemas ocorre, visto que estes tendem a minimizar o erro geral em vez de ter em conta a classe "rara".

Carugo (2007) e Baldi et al. (2000), citados por Martins (2011), referem que podem ser utilizadas métricas estatísticas, com os seguintes valores:

- Positivos Verdadeiros, do inglês *True Positive* (*TP*): casos positivos corretamente previstos.
- Negativos Verdadeiros, do inglês *True Negative* (*TN*): casos negativos corretamente previstos.
- Falsos Positivos, do inglês *False Positive* (*FP*): casos positivos incorretamente previstos.
- Falsos Negativos, do inglês *False Negative* (*FN*): casos negativos incorretamente previstos.

Com estes valores consegue-se ainda criar uma Matriz de Confusão, do inglês *Confusion Matrix*. Esta matriz é constituída por *k* linhas e *k* colunas, consoante o número de classes existentes num certo problema, que se encontra descrita na Tabela 2.2.

Tabela 2.2 – Matriz de Confusão.

		Real	
		A	**B**
Previsto	**A**	TP	FP
	B	FN	TN

Algumas destas métricas estatísticas são descritas por Carugo (2007) e Baldi et al. (2000) e referidas por Martins (2011), tais como:

- *Precision*: é a razão entre os resultados positivos corretamente previstos com todos os resultados classificados como positivos.

$$precision = \frac{TP}{TP + FP} \tag{2.12}$$

- *Recall*: é a razão entre todos os casos reconhecidos como positivos pelo sistema dividido pelo número total de elementos pertencentes à classe positiva.

$$recall = \frac{TP}{TP + FN} \tag{2.13}$$

- *Accuraccy*: fração que representa os casos bem classificados entre os existentes.

$$accuraccy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.14}$$

- *Specificity*: fração dos casos negativos corretamente classificados.

$$specificity = \frac{TN}{TN + FP} \tag{2.15}$$

Através da combinação das expressões descritas anteriormente, podemos construir o *F-measure*, que, como refere Hripcsak & Adam S. Rothschild (2005), é uma média "harmónica" de *Precision* e *Recall*.

$$F = \frac{(1 + \beta^2) \times recall \times precision}{(\beta^2 \times precision) + recall} \tag{2.16}$$

onde *β* permite que se aumente o peso do *Precision* ou do *Recall*, encontrando-se equilibrados quando *β=1*. Em muitos problemas não existe necessidade de aumentar o peso de modo a favorecer o *Precision* ou o *Recall*, resultando, com *β=1*, em:

$$F = \frac{2 \times recall \times precision}{precision + recall} \tag{2.17}$$

2.5 Síntese do capítulo

Neste capítulo foi feita a revisão da literatura, descrevendo diversos algoritmos de *Machine Learning* e os diversos estudos já realizados na área.

Recorrendo a alguns autores, foi ainda descrito o modo de avaliação de um modelo através da enumeração das métricas mais usadas na obtenção de erro.

Foi ainda feita a descrição dos métodos para validar a capacidade de um modelo na perspetiva de alguns autores.

Por último, explicitaram-se alguns trabalhos já realizados relacionados com o tema em estudo nesta dissertação.

No próximo capítulo, serão apresentados os dados de estudo, a metodologia utilizada, a implementação dos algoritmos de *Machine Learning* e a respetiva avaliação. Será ainda referido o modo de seleção da melhor fase da instalação trifásica.

3 Dados e Metodologia

Apresenta-se, neste capítulo, a metodologia aplicada na elaboração deste trabalho, explicitando pormenorizadamente os passos seguidos e a justificação das opções tomadas.

A organização deste capítulo consiste em apresentar o modelo de dados utilizado, realçando aqueles que foram objeto de análise, e descrever os algoritmos usados no projeto.

Depois de terem sido efetuados bastantes estudos nesta área, pode-se afirmar que não existe um único modelo de *Machine Learning* que seja mais correto para cada problema de previsão de energia. Contudo, alguns métodos presentes podem efetuar melhores previsões que outros, para diferentes *data-sets*.

Para o algoritmo ser preciso e flexível é necessário estudar diversas metodologias aplicadas a diferentes amostras de dados. Neste capítulo, discute-se os diferentes métodos utilizados neste projeto e o seu modo de utilização.

O trabalho a desenvolver tem como objetivo dar resposta às perguntas efetuadas anteriormente. De modo a obter as respostas, será adotada uma metodologia de estudo de cariz quantitativa, utilizando séries temporais de forma a obter previsões futuras. Assim, e tendo em conta os tipos de aprendizagem, conclui-se que é supervisionada, sendo o tipo de problema uma Regressão.

Relativamente ao objetivo da investigação, este estudo surge como um caso de uma pesquisa exploratória, no sentido de ser utilizado para possíveis investigações futuras.

3.1 Dados de estudo

A técnica de recolha de dados históricos das instalações trifásicas usadas neste trabalho foi efetuada através de uma parceria entre a empresa *Withus* e a Universidade de Aveiro.

Através da empresa *Withus*, foi possível a recolha de dados históricos de consumo de energia, em formato *Comma-Separated Values* (*CSV*), com um intervalo de tempo de, aproximadamente, três meses e com envio de valores de energia, em *watt-hora* (*Wh*), a cada 15 minutos (uma vez que essa é a frequência de envio de valores de energia), relativos às diferentes fases de instalações trifásicas.

Os conjuntos de dados de consumo de energia fornecidos caracterizam-se por séries temporais, umas com intervalos de tempo contínuos, ou seja, quando são reportados valores de energia regularmente, e outras com intervalos de tempo descontínuos, isto é, quando existem intervalos de tempo onde não é reportado nenhum valor por parte do dispositivo.

O período com dados contínuos decorre entre os dias 28 de julho e 25 de outubro de 2019. Para os dados descontínuos, os registos de consumo situam-se entre os dias 8 de janeiro e 6 de abril de 2020.

3.2 Metodologia

3.2.1 Etapas da metodologia

A metodologia utilizada neste trabalho inclui quatro etapas distintas.

A primeira etapa passa pela criação de duas novas séries temporais, a partir dos dados fornecidos. Uma das séries temporais contém valores periódicos através de funções sinusoidais e a outra contém, somente, o maior intervalo de tempo contínuo extraído da amostra de dados descontínuos, resultando numa amostra com intervalo de tempo de, aproximadamente, seis dias.

A série temporal periódica apresenta o mesmo intervalo de tempo dos dados contínuos, isto é, entre os dias 28 de julho e 25 de outubro de 2019.

Para o caso da série temporal contínua, extraída dos dados descontínuos, o intervalo de tempo é bastante menor, entre os dias 12 e 17 março de 2020.

Para os dados descontínuos, o período de registos situa-se entre os dias 8 de janeiro e 6 de abril de 2020.

Os dados periódicos consideram uma rede hipotética com um padrão de consumo de energia que se repete ao longo do tempo. Este padrão tem um valor mínimo durante a noite e dois valores máximos durante o dia. Com base na fórmula definida por Antunes (2017), a função *Q(t)* foi utilizada para criar a amostra de dados periódicos e apresenta-se da seguinte forma:

$$Q(t) = 150 + \frac{f1(t) + f2(t) + f3(t)}{3} \tag{3.1}$$

onde

$$f1(t) = 100 \times \sin\left(-3 + 2 \times \frac{\pi}{48} \times t\right) \tag{3.2}$$

$$f2(t) = 100 \times \sin\left(-3 + 2 \times \frac{\pi}{96} \times t\right) \tag{3.3}$$

$$f3(t) = 100 \times \sin\left(-11 + 2 \times \frac{\pi}{96} \times t\right) \tag{3.4}$$

e as funções *f1(t)*, *f2(t)* e *f3(t)* modelam o comportamento do utilizador no dia a dia.

O passo seguinte consiste no treino dos diversos algoritmos de *Machine Learning*, para cada série temporal, já mencionada anteriormente. Para que esta etapa seja concretizada, é necessário, em primeiro lugar, dividir o conjunto de dados em dois subconjuntos diferentes: 80% para treino e 20% para teste. Os conjuntos de treino são submetidos aos seis algoritmos.

A terceira etapa inclui a avaliação dos modelos de previsão, através de diferentes métricas de erro: *RMSE*, *MSE*, *MAE*, *MAPE* e R^2, para os conjuntos de teste. A escolha destas métricas deve-se ao facto de serem as mais referidas e utilizadas para problemas de regressão.

No último passo, o algoritmo que apresentar menores valores das métricas de erro será, então, escolhido para selecionar a melhor fase, tendo em conta as diferentes previsões de cada fase de um sistema trifásico.

Apresenta-se, de seguida, na Figura 3.1, a metodologia do trabalho desenvolvido.

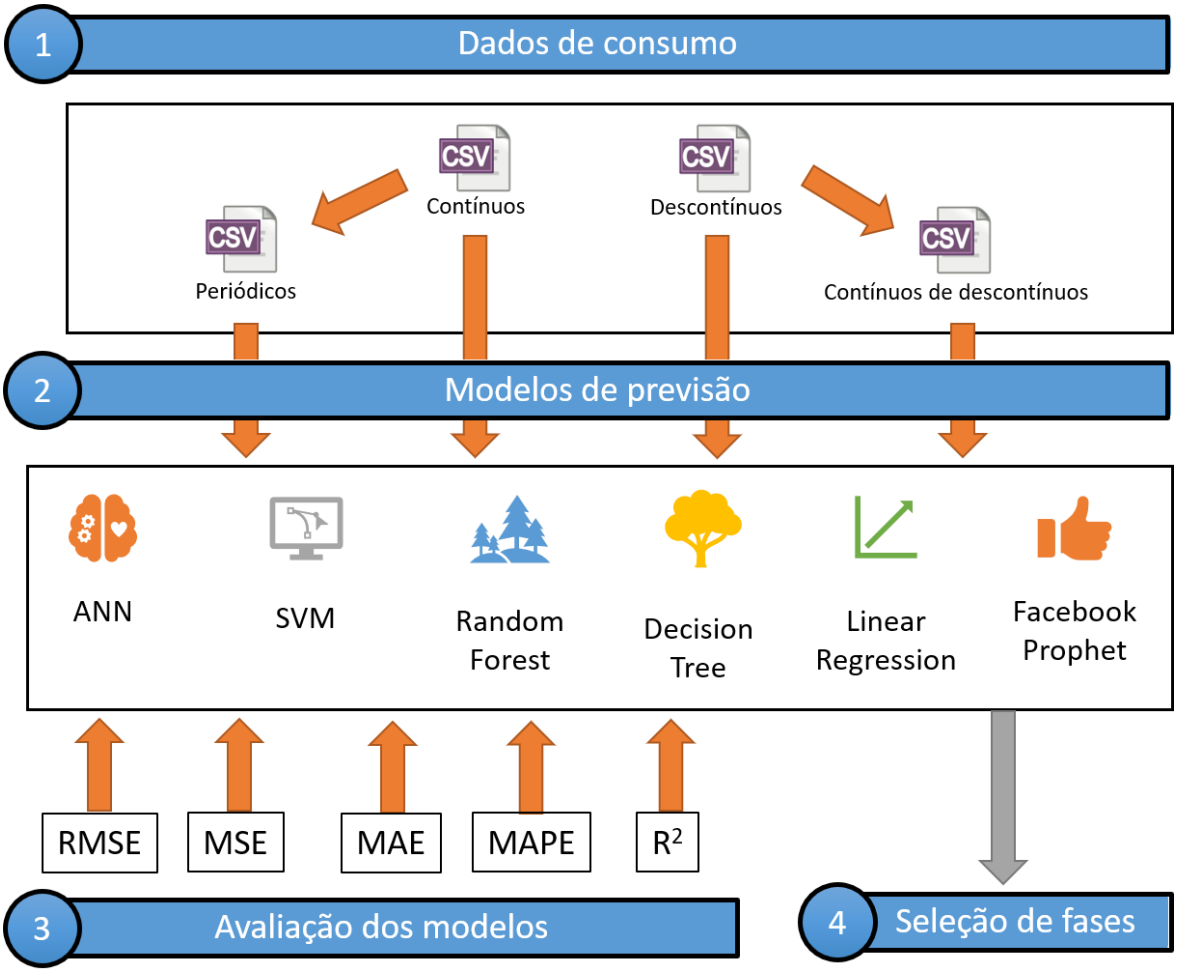

Figura 3.1 – Diagrama das etapas da metodologia do trabalho desenvolvido.

3.2.2 Implementação e avaliação dos modelos

Nem todas as técnicas são as melhores para cada sistema. Por isso, não é espectável encontrar uma solução ideal que se ajuste perfeitamente em todos os tipos de *data-sets*, nem encontrar a melhor solução para cada algoritmo.

A previsão de consumo de energia pode ser vista como um problema de regressão, no qual se inserem alguns dos algoritmos de *Machine Learning* referidos no capítulo anterior. Assim, os modelos usados para resolver o problema foram os seguintes: *SVM*, Redes Neuronais, *Linear Regression, Random Forest, Decision Tree* e *Facebook Prophet*[9].

Estes algoritmos foram escolhidos, porque são os mais viáveis para a modelação de modelos preditivos, justificada pelo consenso da comunidade científica. É de salientar que existe um número significativo de publicações sobre os modelos de previsão, usando os algoritmos referidos.

Para implementar os cinco primeiros modelos neste trabalho, foi utilizada a biblioteca *Scikit-learn* para *Python 3* [10]. Esta permite a criação de modelos de *Machine Learning* de uma forma simples e eficaz. Para o caso do *Facebook Prophet*, usou-se a ferramenta *open source fbprophet*, já disponível para esta linguagem de programação.

Foram usadas bibliotecas do ambiente *SciPy*[11] que ajudam na manipulação de dados (*NumPy*[12] e *pandas*[13]) e na visualização de resultados (*Matplotlib*[14]).

A implementação foi feita da seguinte maneira:

- Carregamento dos Dados: o conjunto de dados foi carregado numa matriz com dimensão equivalente ao número de instâncias existentes nesse conjunto.
- Separação dos Dados: foi necessário dividir a amostra de dados em conjuntos de treino (80%) e de teste (20%). Esta separação dos dados foi feita através do uso da função *train_test_split* do modelo de seleção da biblioteca *sklearn*[15] [16].
- Avaliação do Modelo: as métricas de erro foram calculadas com os valores do conjunto de dados para teste originais.

Para efetuar a validação da previsão, foi usado o *Cross-Validation*, que faz parte do *scikit-learn* e do *fbprophet* em *Python 3*, o que tornou mais fácil a sua implementação. O *adtk*[17], sendo uma biblioteca para *Python 3* que permite a efetuar o *Cross-Validation*, não foi utilizado pelo facto de a biblioteca *sklearn* e o *Facebook Prophet* já conterem *Cross-Validation*.

No final, realizou-se a previsão dos dados de treino e de teste dos modelos, incluindo também funções de avaliação das métricas de *RMSE*, *MSE*, *MAE*, *MAPE* e R^2. No entanto, para a comparação de modelos entre si, utilizaram-se as métricas *RMSE*, *MAPE* e R^2, por serem as mais utilizadas, frequentemente, por diversos autores.

Os resultados serão analisados e discutidos no Capítulo quatro.

O *Scikit* tem como objetivo otimizar o *RMSE* de entre os resultados observados e estimados do modelo, quando se faz o ajuste do mesmo. Esta métrica é utilizada pelo programa, quando se produz previsões para comparar a precisão dos resultados.

O código usado para a implementação dos diferentes algoritmos foi adaptado de *Jason Brownlee*[18] e encontra-se localizado no *GitHub*[19].

Relativamente ao *Support Vector Machine*, foi usada uma versão de *SVM* em *Python 3*, através da biblioteca *sklearn*, mais especificamente, *sklearn.svm.SVR*.

No que concerne à Rede Neuronal, esta é, essencialmente, definida pela forma da própria rede (número de neurónios e de camadas), a função de ativação e algoritmo de otimização. Foram usadas, neste método, seis configurações combinadas com duas funções de ativação (*ReLU* e *Sigmoid*), uma sem ativação, e dois algoritmos de otimização (*Adam* e *SGD*). De modo a realizar previsões de Redes Neuronais com organização linear, o uso de *Keras*[20] [21] do *TensorFlow 2.0* facilita o ajuste, avaliação e predição deste modelo.

Em relação ao *Linear Regression*, a complexidade deste algoritmo depende do número de variáveis independentes introduzidas. Neste caso, usou-se o *Simple Linear Regression*, visto que só existe uma variável independente, sendo esta a de energia consumida.

No caso do *Decision Tree*, são utilizados modelos em árvore para determinar as suas decisões e possíveis resultados de eventos. Foi usada a biblioteca *tree*[22] do *sklearn* para alcançar a previsão através deste algoritmo.

No que diz respeito ao *Random Forest*, este pode ser modelado pelo número de árvores na floresta aleatória e o número mínimo de amostras requeridas para cada divisão. Como este estudo assume o carácter de regressão, utilizou-se o *Random Forest Regression*[23].

Por último, fez-se uso do *Facebook Prophet*[24], usando a biblioteca *fbprophet* para *Python 3*. É um procedimento para prever dados de séries temporais, no qual as tendências não lineares se ajustam à sazonalidade anual, semanal e diária, além de efeitos de férias. Contudo, neste projeto, foi efetuada uma previsão que se ajusta somente à sazonalidade semanal e mensal. Foram escolhidos 30 dias para fazer a predição, com uma frequência de 15 minutos.

Para uma melhor visualização, apresentam-se, na Tabela 3.1, os algoritmos implementados, bem como as respetivas configurações.

Tabela 3.1 – Algoritmos aplicados e respetivas configurações.

Algoritmos	***Configurações***
ANN	*optimizer=Adam*
	optimizer=SGD
	activation=ReLU, *optimizer=Adam*
	activation=ReLU, *optimizer=SGD*
	activation=Sigmoid, *optimizer=Adam*
	activation=Sigmoid, *optimizer=SGD*
SVR	---
Decision Tree	---
Linear Regression	---
Random Forest	---
Facebook Prophet	periods=30, freq='15min'

3.2.3 Seleção de fases

Após a avaliação dos algoritmos e a escolha daquele que apresenta melhores previsões, avança-se, então, para a construção de um outro programa, em *Python*

3, com o objetivo de escolher a melhor fase de um sistema trifásico, na qual se deve injetar a energia produzida por painéis fotovoltaicos, em função das previsões de cada fase.

Assim, torna-se necessário a existência uma forma de comunicação entre o programa principal e o equipamento de medição de consumo energético, de modo a permitir o envio de séries temporais de consumo energético remotamente. Para tal, utiliza-se a ferramenta de programação *Sockets*[25] para *Python 3*, resultando num programa-cliente que envia três *data-sets* simultaneamente e num programa-servidor que realiza o processamento a cada três conjuntos de dados recebidos. No final, o programa-servidor envia uma informação para a instalação trifásica, identificando a fase para a qual deve comutar, que será a que tiver um maior consumo de energia.

A Figura 3.2 apresenta a estrutura da implementação do projeto de dissertação.

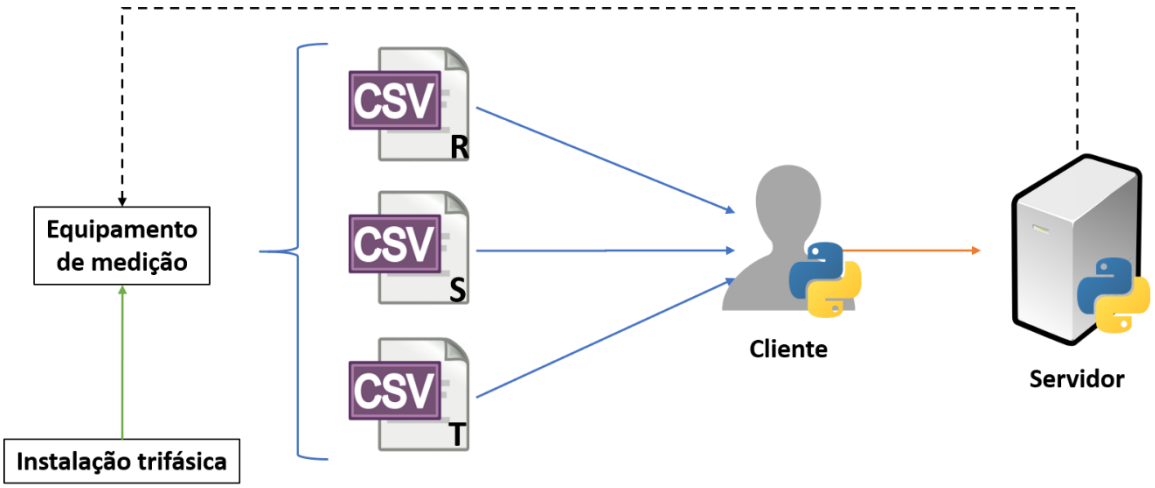

Figura 3.2 – Esquema teórico da comunicação do projeto.

3.3 Síntese do capítulo

Neste capítulo, foram caracterizados os dados de consumo real utilizados neste estudo.

Explicitaram-se ainda as etapas da metodologia usada, sendo também descrito, não só o modo de implementação dos modelos, mas também a avaliação dos mesmos. Descreveram-se as métricas de erro utilizadas, as configurações usadas para cada algoritmo e o modo de funcionamento na escolha de fases.

No capítulo seguinte, é feita uma análise da visualização gráfica dos resultados obtidos pela utilização dos algoritmos anteriormente referidos e uma síntese dos mesmos para cada série temporal.

4 Resultados e discussão

Neste capítulo, são apresentados e discutidos os resultados dos algoritmos selecionados com base na metodologia exposta no capítulo anterior. Primeiro, são analisados e discutidos os perfis de consumo e de previsão de energia elétrica num dado espaço temporal. De seguida, são analisados e comparados os resultados da previsão após aplicação dos algoritmos selecionados nesta dissertação.

Após a análise dos resultados dos modelos, devemos utilizar, na escolha de fases de uma instalação trifásica, o algoritmo que apresentar mais vantagens.

Este processo pode ser realizado executando o algoritmo, utilizando *data-sets* conhecidos ou gerados, quer com dados reais encontrados nas máquinas que reportam esses valores, quer através de dados periódicos. Ao comparar os resultados dos algoritmos apresentados entre si, podemos verificar a sua viabilidade e qualidade dos mesmos na previsão.

Os modelos previamente descritos (*ANN*, *SVM*, *Decision Tree*, *Random Forest, Linear Regression* e *Facebook Prophet*) foram aplicados a quatro conjuntos de dados (periódicos, contínuos, descontínuos e contínuos provenientes de descontínuos). Todos estes *data-sets* têm um formato idêntico entre eles, ou seja, apresentam valores reais, num período temporal de, aproximadamente, três meses (exceto o último *data-set*, em que a dimensão é variável consoante o número de valores no maior intervalo de tempo contínuo e, neste caso, este conjunto de dados tem a duração de, aproximadamente, seis dias), e com instâncias de valor a cada 15 minutos.

Para se poder realizar a previsão, foi criado um programa em *Python 3*. Este programa realiza previsões com base no algoritmo e no conjunto de dados definidos *a priori*, calculando e apresentando os resultados das métricas utilizadas.

Os gráficos representados entre a Figura 4.1 e a Figura 4.77 apresentam os dados originais (cor azul), a previsão de treino (cor-de-laranja) e de teste (cor verde) e relacionam o consumo de energia, de acordo com o intervalo de tempo definido. Estes gráficos são apresentados dois a dois, (exceto para os dados contínuos provenientes de descontínuos, em que é apresentado apenas um gráfico para cada algoritmo), sendo que, um gráfico que apresenta o intervalo de tempo máximo do

conjunto de dados (três meses ou seis dias, consoante cada *data-set*) e o outro um intervalo de tempo de 48 ou 72 horas. A situação ideal é aquela em que a diferença entre os valores reais e os previstos é a menor possível.

Na discussão dos resultados será utilizado o conceito de amplitude como a diferença entre o momento de maior consumo (pico) e o menor consumo.

Para avaliar os modelos, efetuaram-se dois passos: o primeiro consistiu na avaliação de cada amostra de dados e no segundo passo calculou-se a respetiva métrica, tendo em conta o conjunto de dados para teste.

As métricas usadas foram o *Mean Error*, *Mean Absolute Error*, *Root Mean Square Error*, *Mean Absolute Percentage Error* e o Coeficiente de Determinação, mas só os *RMSE* (usado pelas bibliotecas *Scikit* no processo de ajuste), *MAPE* e R^2 são consideradas as melhores na comparação dos algoritmos, pelas razões já plasmadas no capítulo anterior.

Para avaliar o desempenho de cada método, os resultados obtidos pela aplicação de cada modelo foram comparados entre si, considerando as métricas descritas anteriormente.

4.1 Dados periódicos

As figuras com numeração ímpar apresentam uma previsão com um intervalo de tempo entre o dia 28 de julho e o dia 25 de outubro de 2019.

Para uma melhor perceção do perfil do padrão de consumo de energia, as figuras pares apresentam o consumo real e a previsão de apenas dois dias (8 e 9 de outubro de 2019) do período atrás mencionado, pertencente ao início do conjunto de teste.

Em relação ao *ANN*, sem função de ativação e com optimizador *Adam*, na Figura 4.1, verifica-se que a previsão apresenta uma amplitude inferior aos dados reais. Na Figura 4.2, constata-se que a série temporal prevista acompanha a amostra dos dados reais, ou seja, apresenta o mesmo perfil de padrão do consumo de energia elétrica.

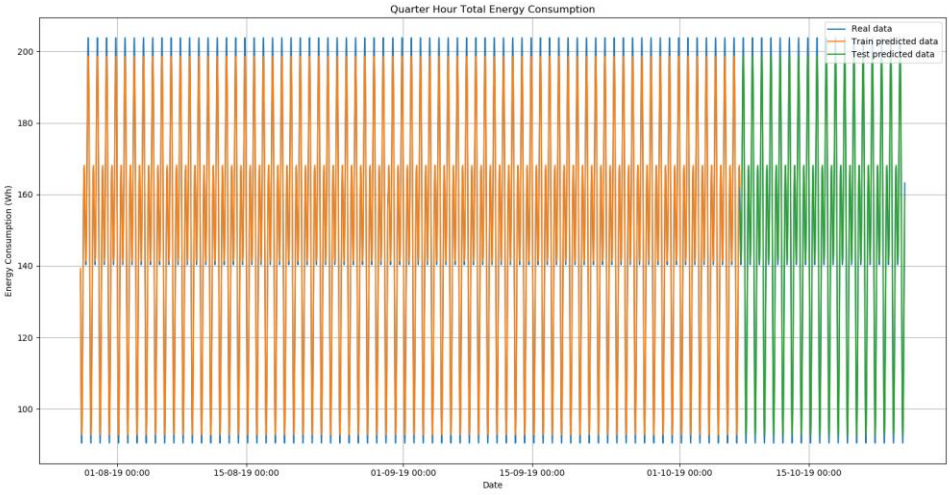

Figura 4.1 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *ANN*, optimizador *Adam*.

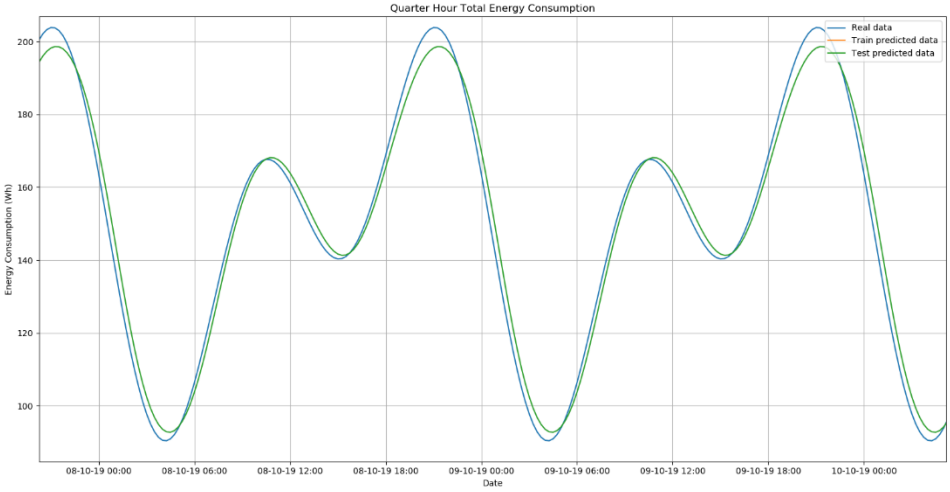

Figura 4.2 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *ANN*, optimizador *Adam*.

No que concerne ao *ANN,* com ativação *ReLU* e com optimizador *Adam*, verifica-se que, na Figura 4.3, tal como na situação anterior, os resultados são similares. Na Figura 4.4 verifica-se que a previsão também acompanha o padrão do consumo de energia elétrica, com uma amplitude ligeiramente inferior à real.

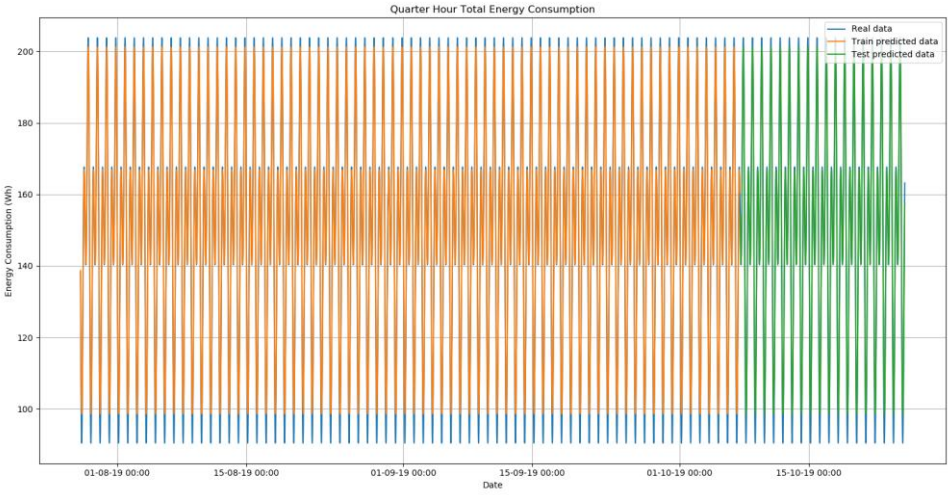

Figura 4.3 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

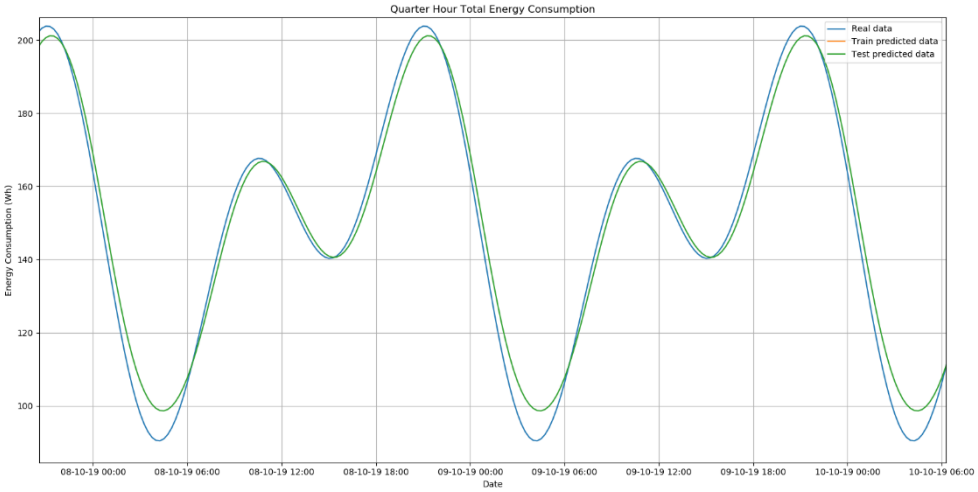

Figura 4.4 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

Relativamente ao *ANN*, com optimizador *SDG* e função de ativação *ReLU*, observa-se que, na Figura 4.5 e na Figura 4.6, a amplitude da previsão é bastante inferior à das anteriores, resultando em previsões mais desfavoráveis.

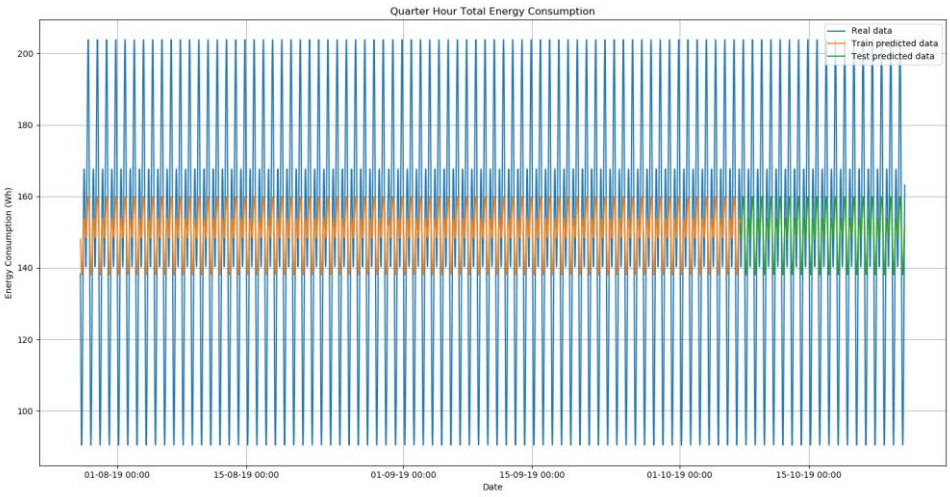

Figura 4.5 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

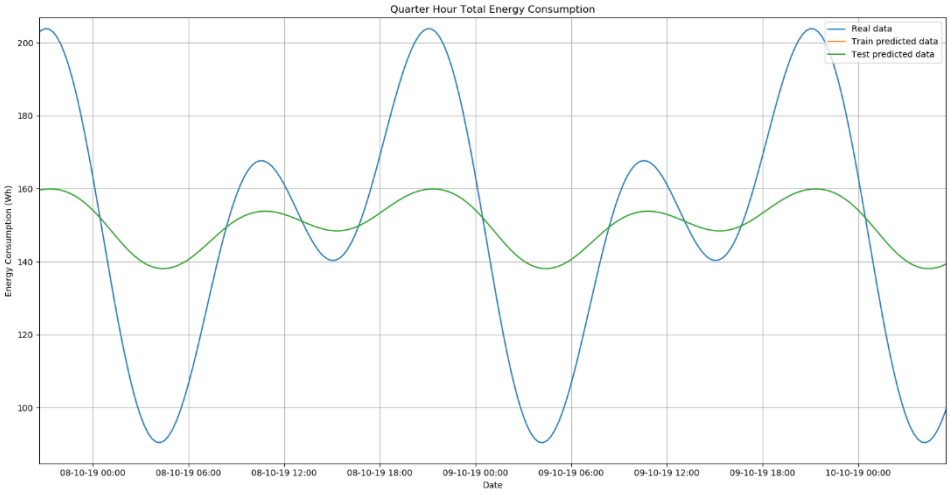

Figura 4.6 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

Em relação ao *ANN*, com o optimizador *SGD*, verifica-se que, na Figura 4.7 e na Figura 4.8, os gráficos são semelhantes aos anteriores (Figura 4.5 e Figura 4.6), apresentando amplitudes significativamente mais baixas em relação aos valores originais. Tal como a anterior, não se pode considerar esta configuração como solução para este problema.

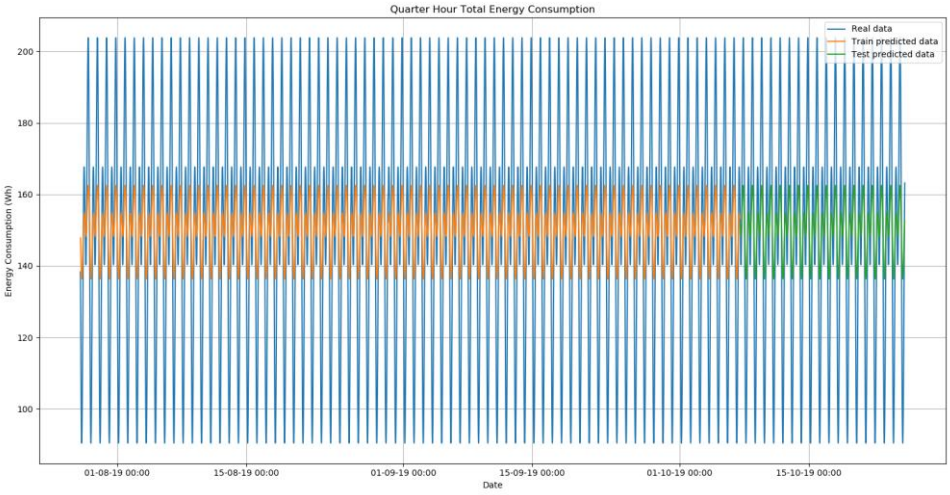

Figura 4.7 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *ANN*, optimizador *SGD*.

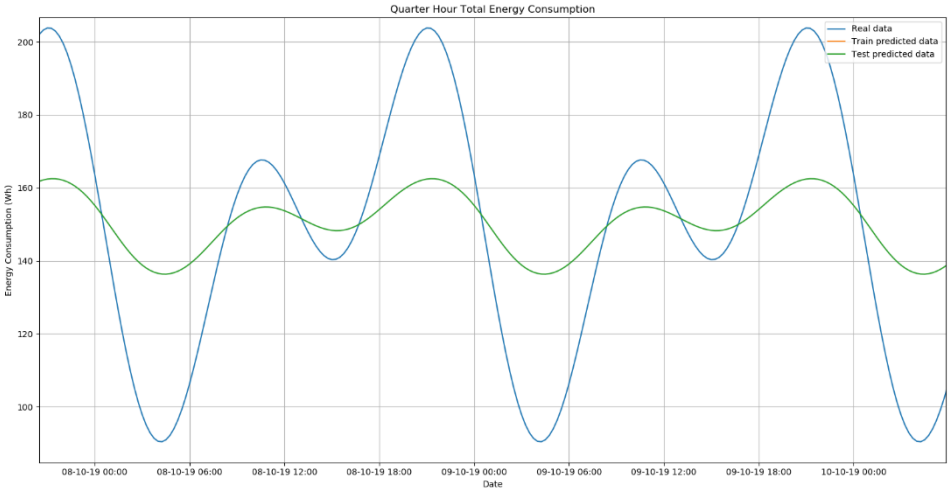

Figura 4.8 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *ANN*, optimizador *SGD*.

No que respeita ao *ANN*, com função de ativação *Sigmoid* e optimizador *Adam*, verifica-se que, na Figura 4.9, a previsão apresenta resultados mais desfavoráveis quando comparados com os que são apresentados pelas configurações de *ANN*, com ativação *ReLU* e sem nenhuma função de ativação.

Ao observar a Figura 4.10, consegue-se identificar, no gráfico, que a previsão apresenta uma amplitude bastante inferior relativamente à série temporal real.

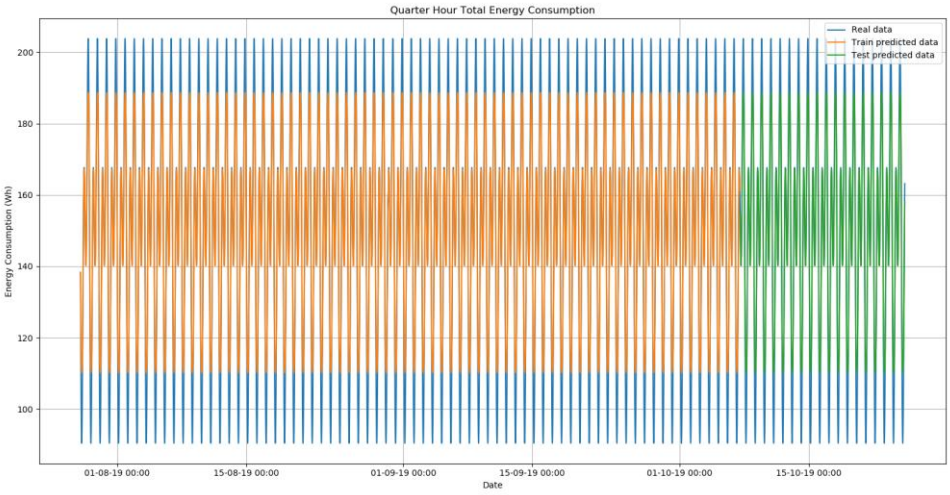

Figura 4.9 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

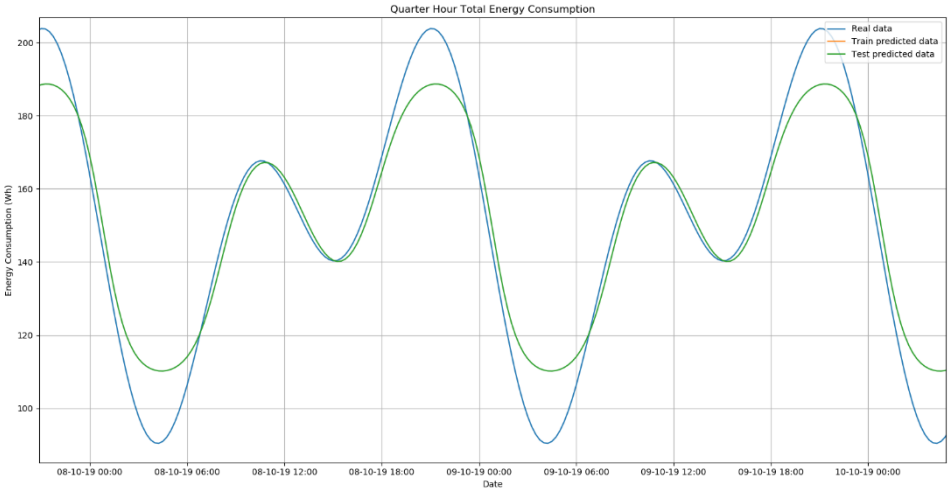

Figura 4.10 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

No caso do *ANN*, com função de ativação *Sigmoid* e optimizador *SGD*, verifica-se que, na Figura 4.11, a amplitude da previsão é muito baixa em comparação com as anteriores. Da mesma forma, na Figura 4.12, o gráfico

apresenta uma previsão de consumo praticamente constante durante os dois dias selecionados e uma amplitude pouco significativa.

Pode-se afirmar que, com base nos gráficos de previsão utilizando *ANN*, as configurações com o optimizador *SGD* são desfavoráveis.

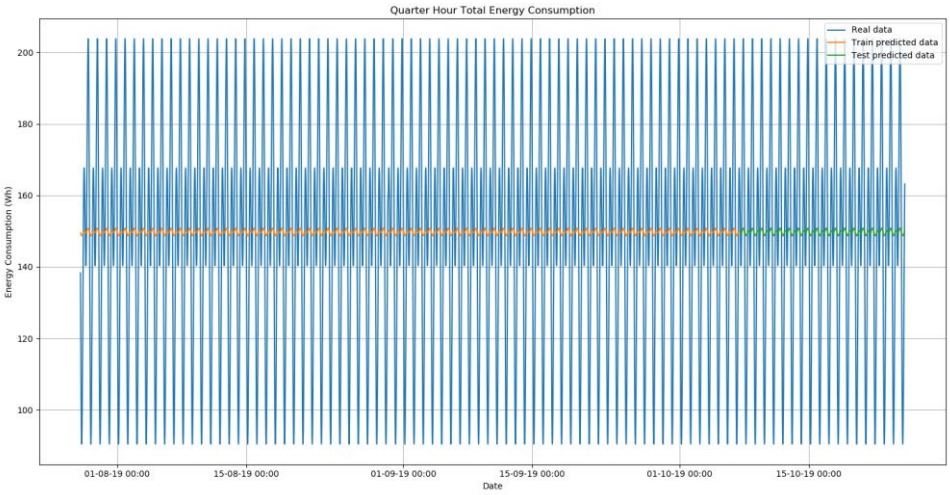

Figura 4.11 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

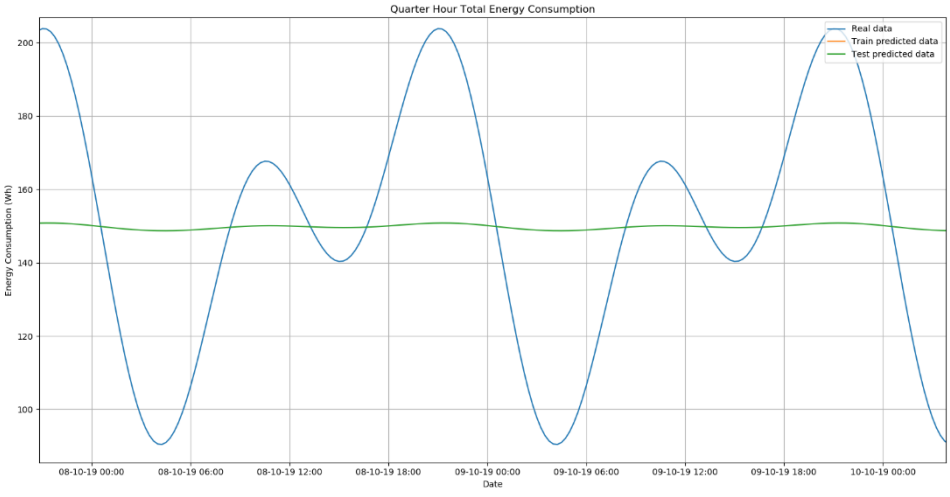

Figura 4.12 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

Relativamente ao *Random Forest*, na Figura 4.13 e na Figura 4.14, consegue-se verificar que a amplitude da previsão é igual à dos dados originais. Observa-se também que não são visíveis diferenças entre os valores originais e os valores previstos. Assim, considera-se que este algoritmo responde às necessidades do problema.

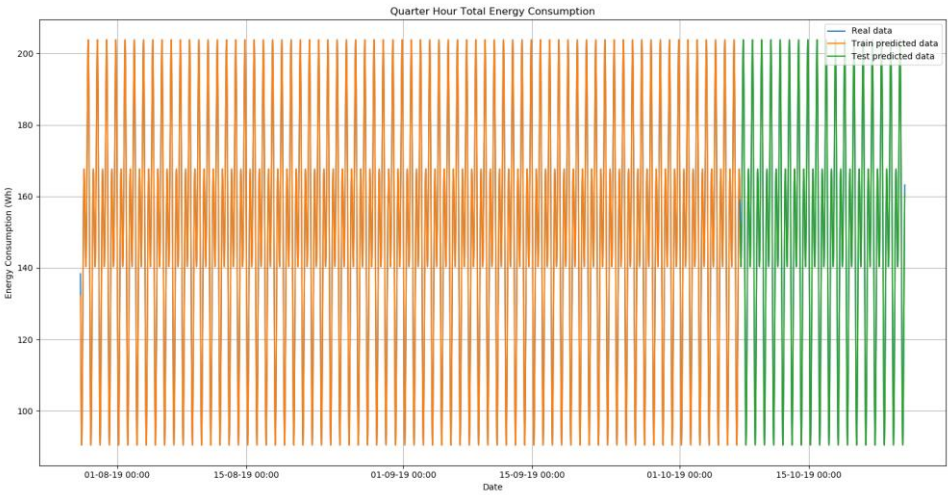

Figura 4.13 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *Random Forest*.

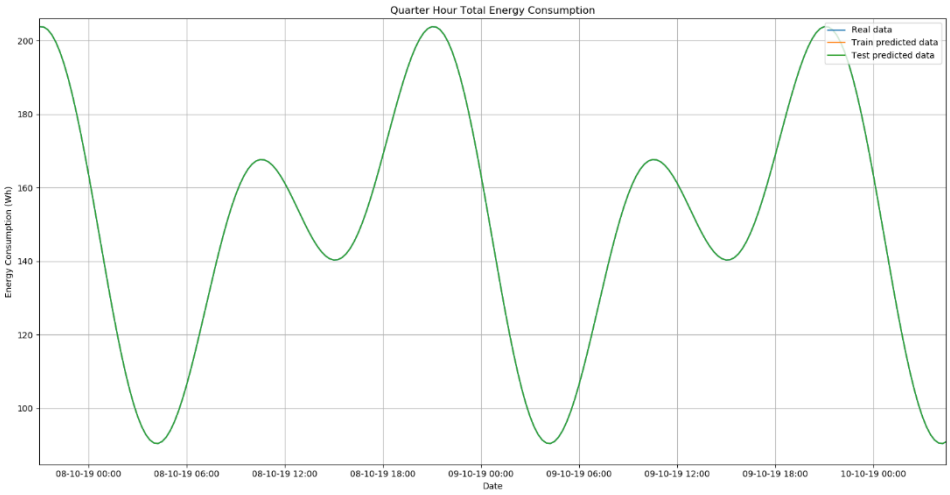

Figura 4.14 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *Random Forest*.

No caso do *Linear Regression*, a Figura 4.15 mostra uma previsão com uma amplitude praticamente igual à dos valores originais. No caso da Figura 4.16, o gráfico apresenta apenas um ligeiro atraso entre o valor da previsão e o valor real. Deste modo, pode-se concluir que este algoritmo apresenta resultados satisfatórios.

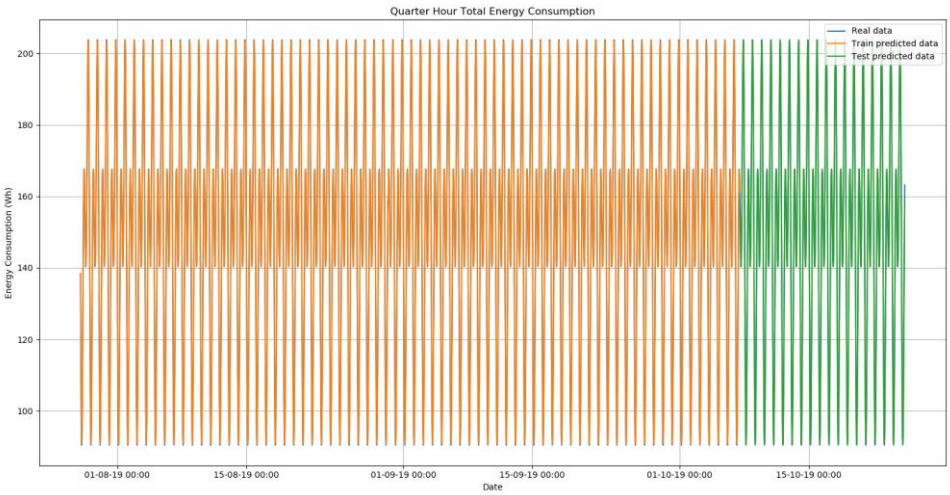

Figura 4.15 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *Linear Regression*.

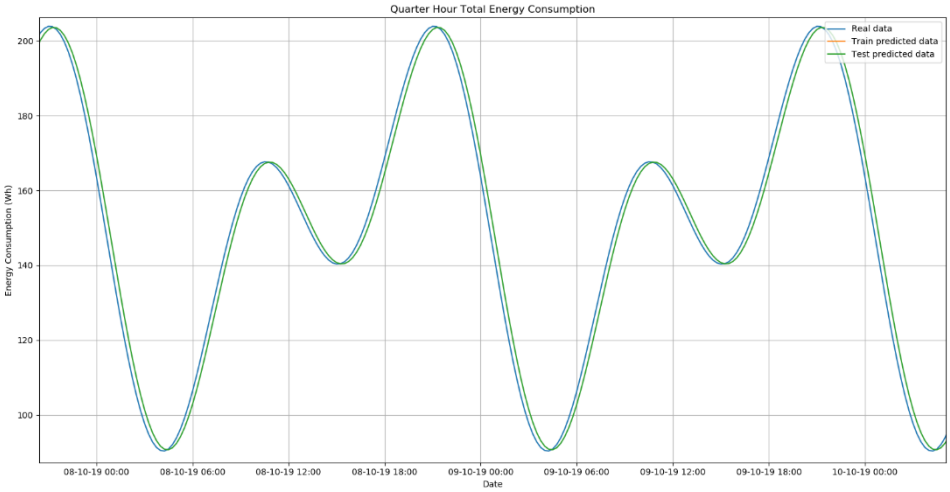

Figura 4.16 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *Linear Regression*.

Relativamente ao *Facebook Prophet*, pela observação unicamente do gráfico da Figura 4.17, não é possível fazer considerações sobre a previsão, em relação aos dados reais. Na Figura 4.18, observa-se que a previsão apresenta um padrão quase idêntico relativo ao dos dados fornecidos, com apenas um ligeiro adiantamento, ou seja, este algoritmo responde de forma satisfatória.

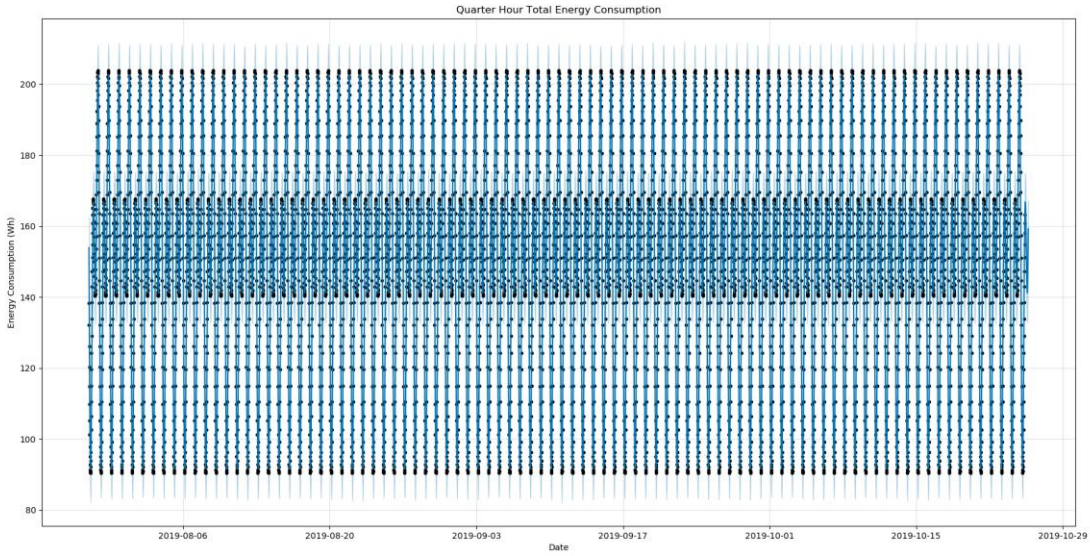

Figura 4.17 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *Facebook Prophet*.

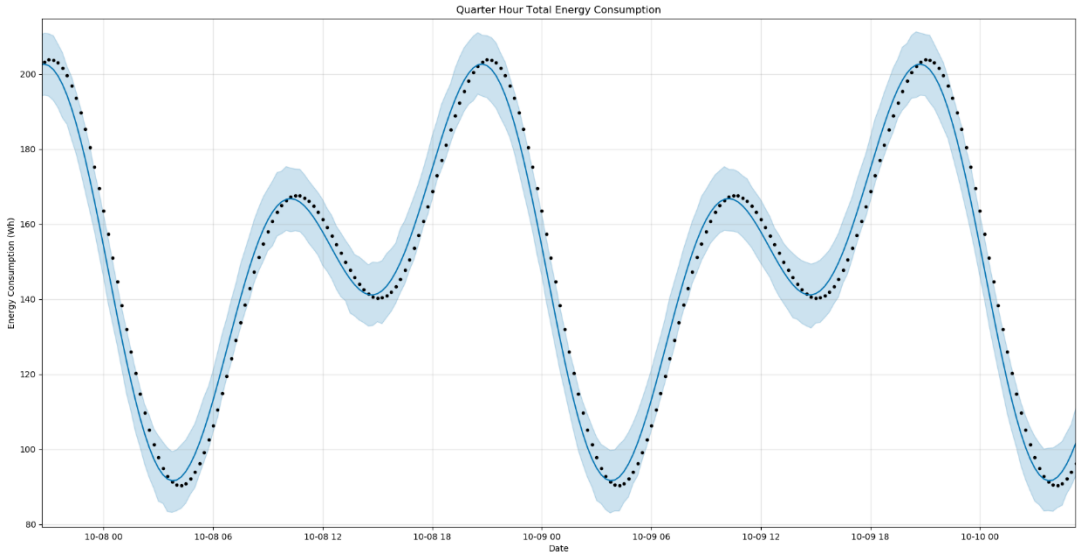

Figura 4.18 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *Facebook Prophet*.

Em relação ao *SVR*, na Figura 4.19, a previsão resulta numa menor amplitude relativamente à dos valores reais. Na Figura 4.20, observa-se, não só um achatamento nos valores mínimos e valores máximos da previsão, quando comparado com a da série temporal original, mas também uma amplitude ligeiramente inferior, tornando este algoritmo numa solução não-ideal.

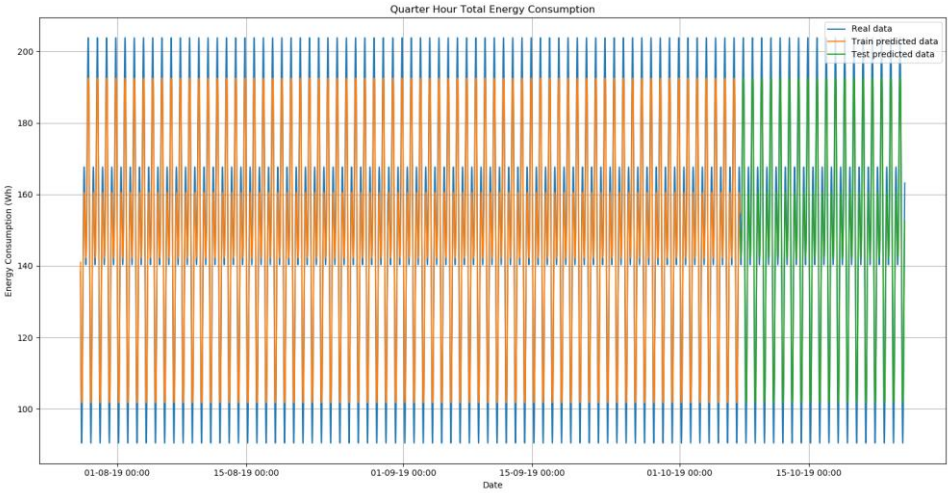

Figura 4.19 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *SVR*.

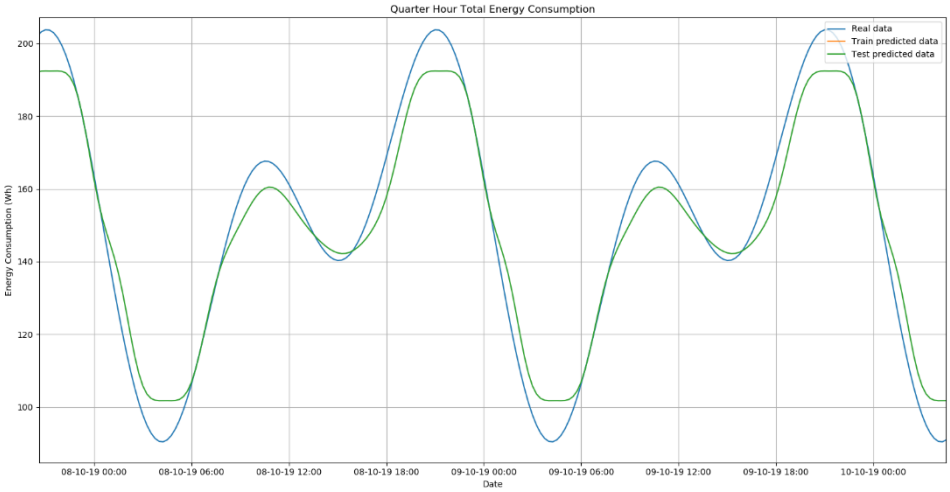

Figura 4.20 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *SVR*.

No que concerne ao *Decision Tree*, observa-se que a Figura 4.21 apresenta uma previsão semelhante à do *Random Forest* em termos de amplitude. Da mesma forma, a Figura 4.22 comprova essa semelhança, na qual é demonstrada, tal como o *Random Forest*, a melhor previsão relativamente aos algoritmos anteriores, isto é, a diferença entre os valores originais e os da previsão é nula.

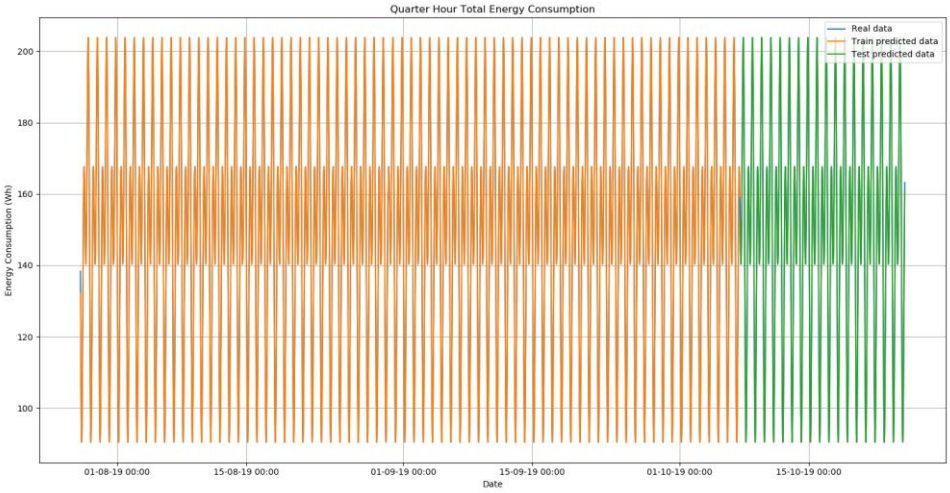

Figura 4.21 – Consumo real e previsão de energia durante três meses, com dados periódicos, modelo *Decision Tree*.

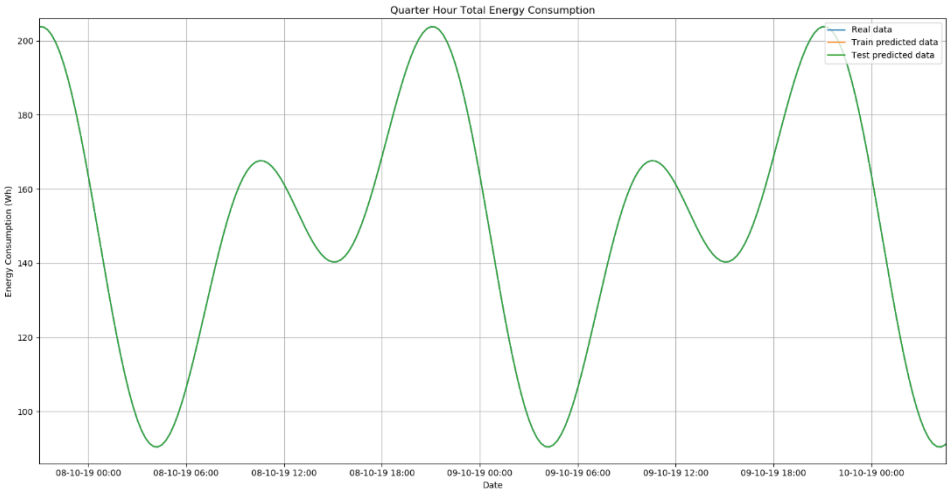

Figura 4.22 – Consumo real e previsão de energia durante dois dias consecutivos, com dados periódicos, modelo *Decision Tree*.

Pela análise dos gráficos apresentados anteriormente, verificamos que a amplitude varia consoante o algoritmo utilizado. De uma forma visual, podemos afirmar que as situações que melhor respondem a esta amostra de dados são os algoritmos *Decision Trees*, *Random Forest* e Regressão Linear.

A Tabela 4.1 apresenta os valores obtidos para cada métrica utilizada neste estudo e confirma o resultado de análise dos gráficos anteriores.

Tabela 4.1 – Síntese dos resultados obtidos após a aplicação dos modelos utilizados para os dados periódicos.

Modelo	*RMSE*	*MSE*	*MAE*	*MAPE* (%)	R^2
***ANN* (Sem Ativação, *Adam*)**	3.66	13.42	3.21	2.25	0.99
***ANN* (Sem Ativação, *SGD*)**	21.46	460.71	17.22	12.91	0.56
***ANN* (*ReLU*, *Adam*)**	5.81	33.8	4.72	3.55	0.97
***ANN* (*ReLU*, *SGD*)**	25.63	656.65	20.58	15.45	0.37
***ANN* (*Sigmoid*, *Adam*)**	7.87	61.86	6.06	4.67	0.94
***ANN* (*Sigmoid*, *SGD*)**	31.75	1007.9	25.53	19.1	0.04
SVR	6.8	46.28	5.58	3.96	0.96
Decision Tree	0	0	0	0	1
Linear Regression	3.41	11.64	2.94	2.07	0.99
Random Forest	0	0	0	0	1
Facebook Prophet	6.54	42.73	5.45	3.85	0.96

Observa-se que os testes realizados, sem considerar ruído, mostram resultados perfeitos (*RMSE*=0, *MAPE*=0 e R^2=1) para os algoritmos *Decision Trees* e *Random Forest*, podendo ser considerados como ideais para este tipo de conjunto de dados. De referir que a Regressão Linear também apresenta resultados satisfatórios.

4.2 Dados contínuos

Este tipo de dados usados é proveniente dos sistemas de consumo de energia, os quais contêm intervalos de tempo de, aproximadamente, três meses e

com valores a cada 15 minutos. Estes são os indicados para serem usados no teste e avaliação, pois incluem os valores de energia consumidos pelo utilizador.

As figuras com numeração ímpar apresentam um intervalo de tempo entre o dia 28 de julho e o dia 25 de outubro de 2019.

Para os gráficos das figuras com numeração par, o consumo real e a previsão correspondem a um intervalo de tempo de dois dias (8 e 9 de outubro de 2019) do período atrás mencionado, pertencente ao início do conjunto de teste.

Com a utilização do *ANN*, sem função de ativação e com optimizador *Adam*, constata-se que, na Figura 4.23, a previsão de consumo apresenta uma amplitude menor relativamente à dos dados originais. Na Figura 4.24, verifica-se que a previsão apresenta um padrão de consumo semelhante ao dos dados reais.

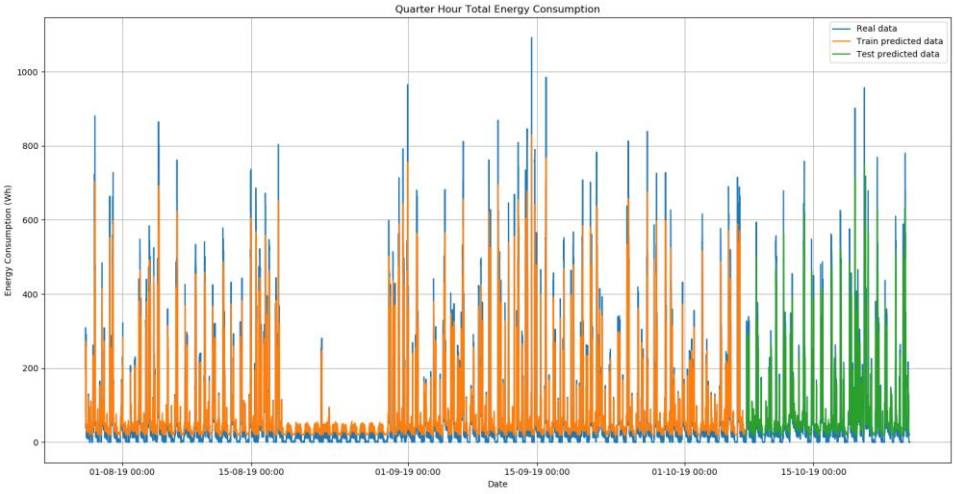

Figura 4.23 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *ANN*, optimizador *Adam*.

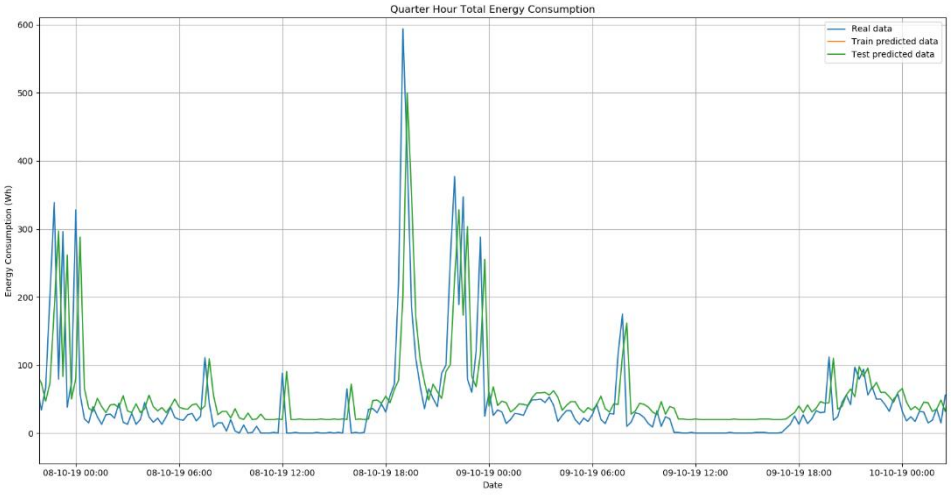

Figura 4.24 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *ANN*, optimizador *Adam*.

Relativamente ao *ANN*, com ativação *ReLU* e com optimizador *Adam*, observa-se que, na Figura 4.25, tal como *ANN* sem função de ativação, a amplitude de previsão é menor em comparação com a série temporal fornecida. Na Figura 4.26 verifica-se que, tal como no caso do *ANN* sem função de ativação, esta previsão demonstra um padrão do consumo de energia elétrica, com uma amplitude ligeiramente inferior ao real.

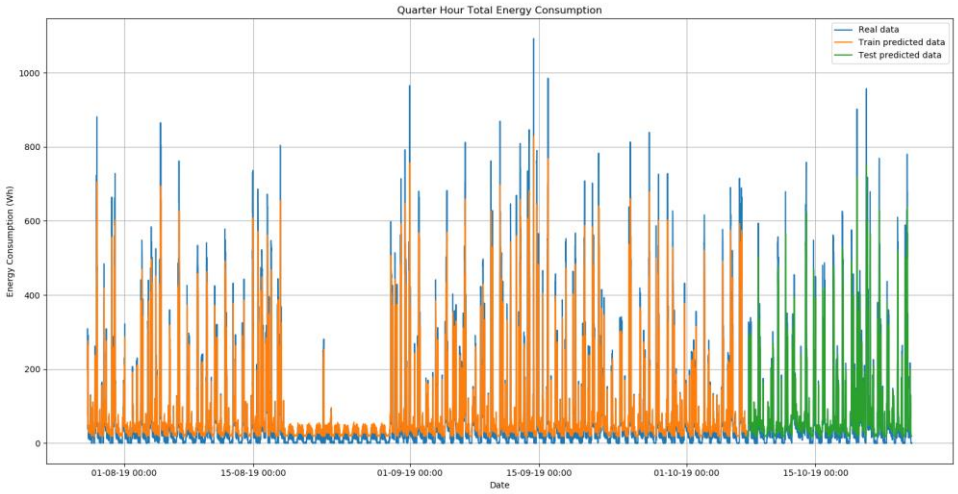

Figura 4.25 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

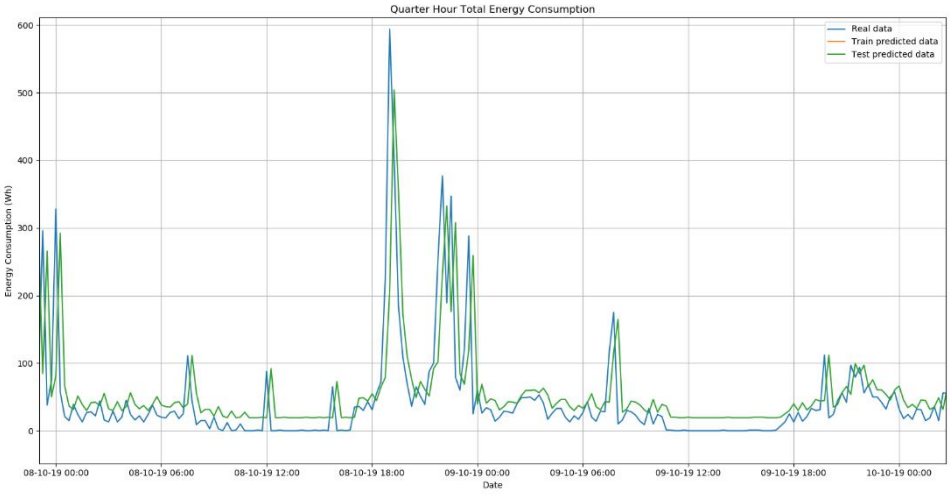

Figura 4.26 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

No caso do *ANN*, com um optimizador *SDG* e função de ativação *ReLU*, observa-se que, na Figura 4.27, a previsão apresenta uma amplitude significativamente inferior à dos valores reais. Constata-se, com mais detalhe, na Figura 4.28, que os valores da previsão são praticamente constantes ao longo do tempo e, por conseguinte, esta configuração não deve ser utilizada.

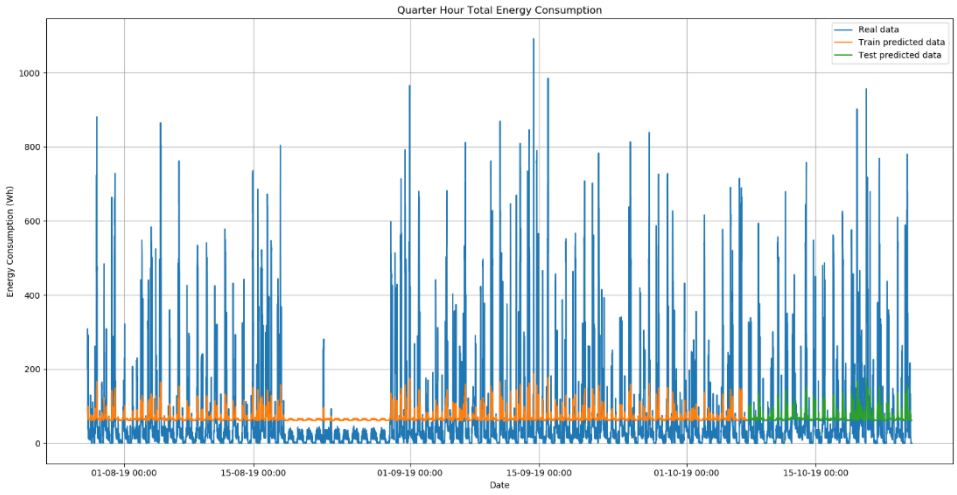

Figura 4.27 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

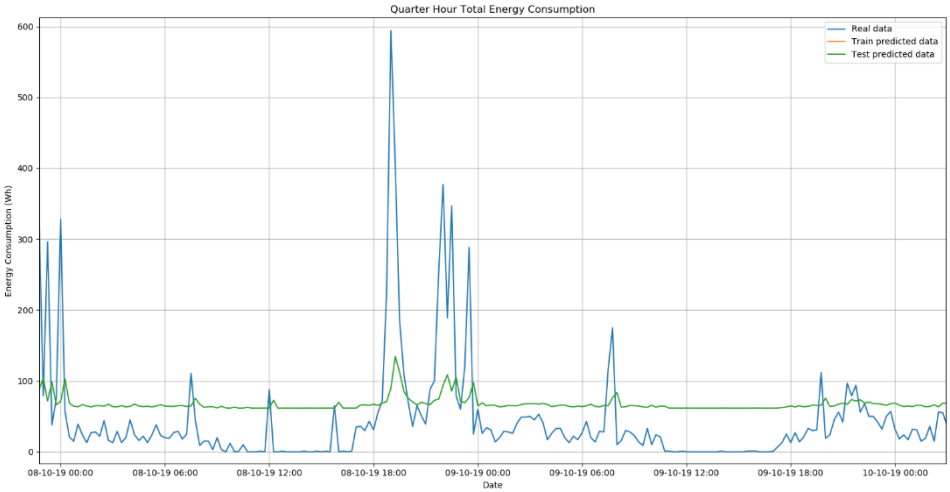

Figura 4.28 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

Com a utilização do *ANN*, com o optimizador *SGD* e sem função de ativação, observa-se, na Figura 4.29 e na Figura 4.30, que a amplitude de previsão é bastante inferior em comparação com a da amostra de dados real. Esta configuração, tal como a anterior, não pode ser considerada como solução para este problema.

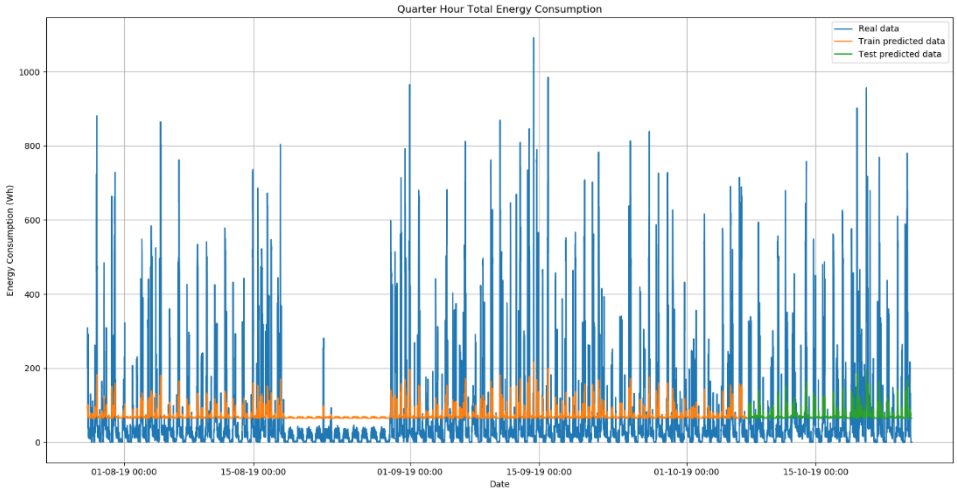

Figura 4.29 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *ANN*, optimizador *SGD*.

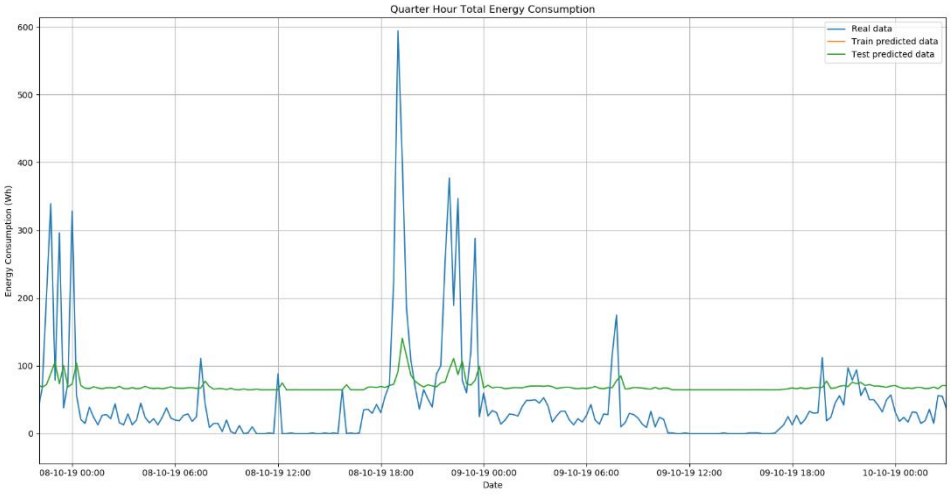

Figura 4.30 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *ANN*, optimizador *SGD*.

No que respeita ao algoritmo *ANN*, com função de ativação *Sigmoid* e optimizador *Adam*, constata-se que, na Figura 4.31, os valores previstos apresentam uma amplitude significativamente inferior à da série temporal original. Também se pode observar, agora com mais detalhe, na Figura 4.32, que os valores mínimos de previsão se situam acima dos valores reais, concluindo que esta configuração é de evitar.

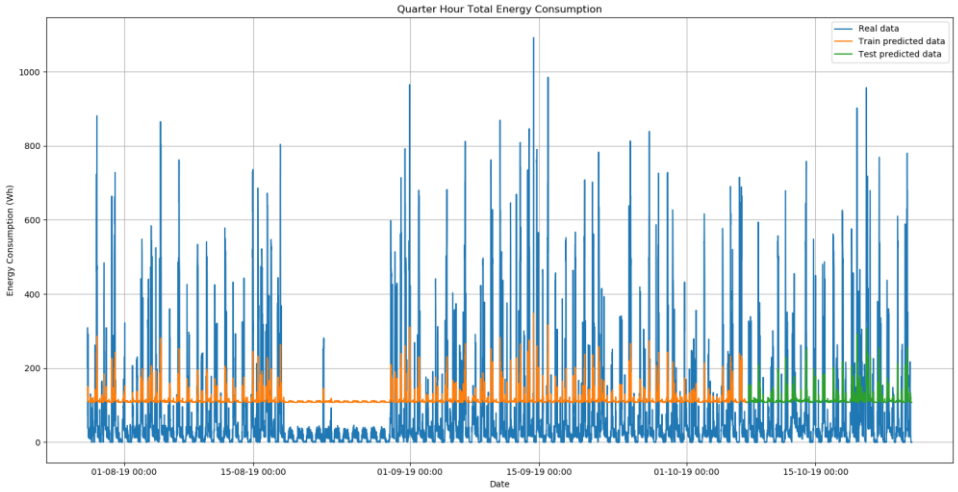

Figura 4.31 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

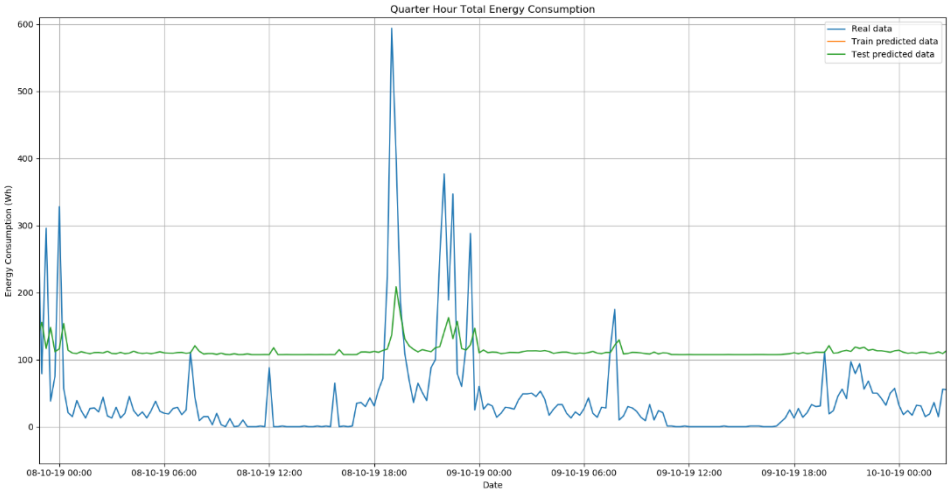

Figura 4.32 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

Relativamente à configuração de *ANN*, com função de ativação *Sigmoid* e optimizador *SGD*, verifica-se que, na Figura 4.33, os valores da previsão são praticamente constantes ao longo do tempo. O mesmo se observa na Figura 4.34, cuja previsão é semelhante às que resultaram da configuração de *ANN* com optimizador *SGD*, tornando desfavorável o uso deste optimizador.

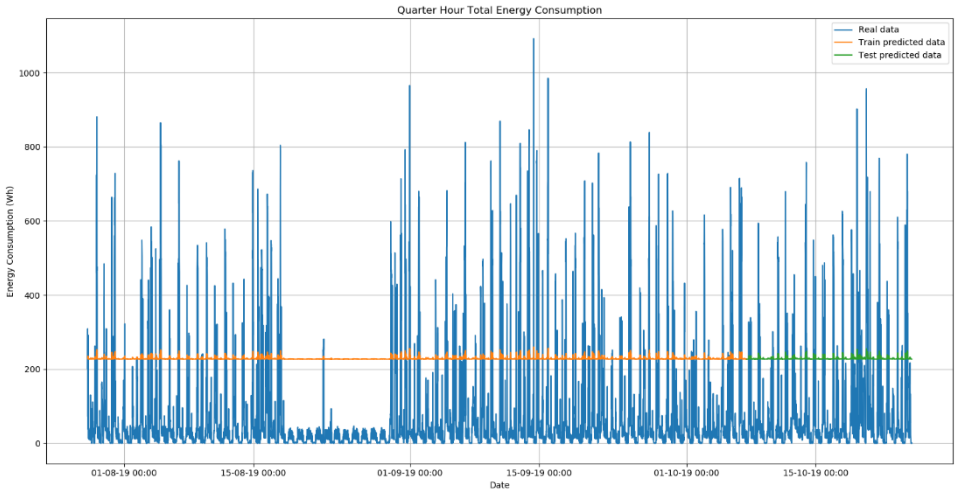

Figura 4.33 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

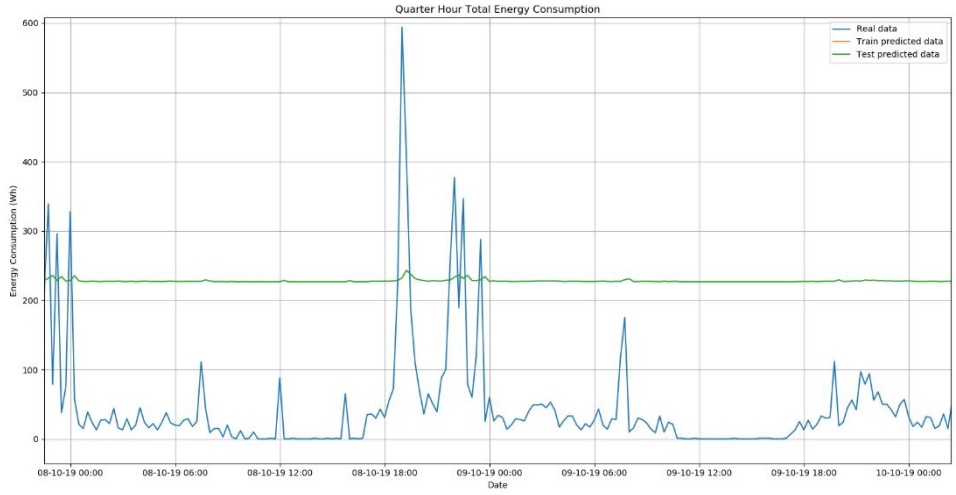

Figura 4.34 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

Em relação ao *Random Forest*, na Figura 4.35, verifica-se que a previsão apresenta uma amplitude ligeiramente inferior comparada com a dos dados fornecidos, incidindo maioritariamente nos valores máximos de consumo. Na Figura 4.36, observa-se, com mais pormenor, que a previsão manteve o padrão da série temporal fornecida. Pela observação dos gráficos, pode-se afirmar que este modelo demonstra uma previsão satisfatória.

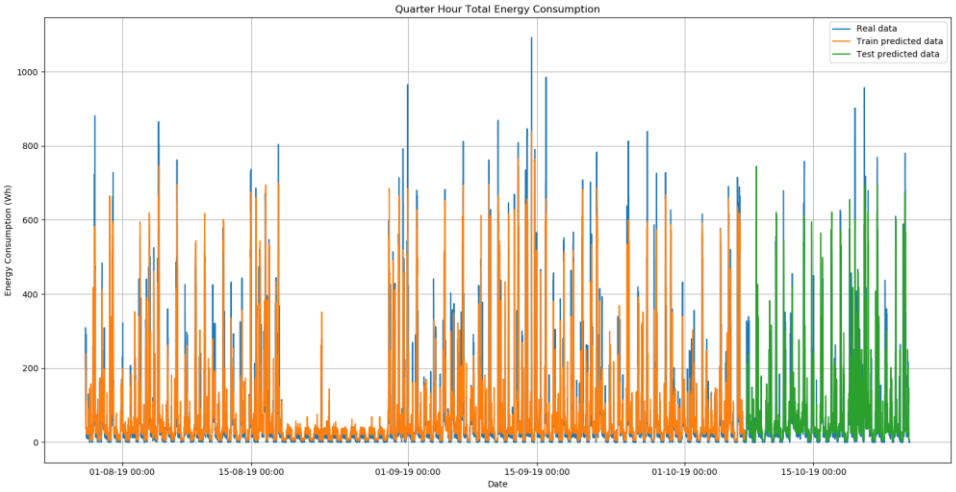

Figura 4.35 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *Random Forest*.

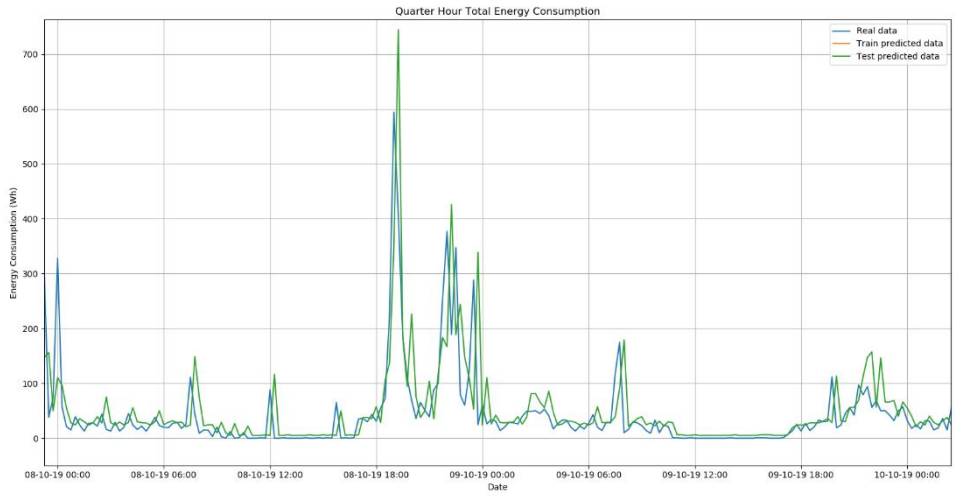

Figura 4.36 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *Random Forest*.

No caso do *Linear Regression*, o gráfico da Figura 4.37, à semelhança do *Random Forest*, apresenta uma previsão semelhante aos valores originais, quer no padrão de consumo, quer na amplitude, evidenciando, no entanto, um valor menor nos picos de consumo. No caso da Figura 4.38, a previsão segue o padrão de perfil de consumo da série temporal real. Assim, considera-se que este algoritmo responde às necessidades de resolução do problema.

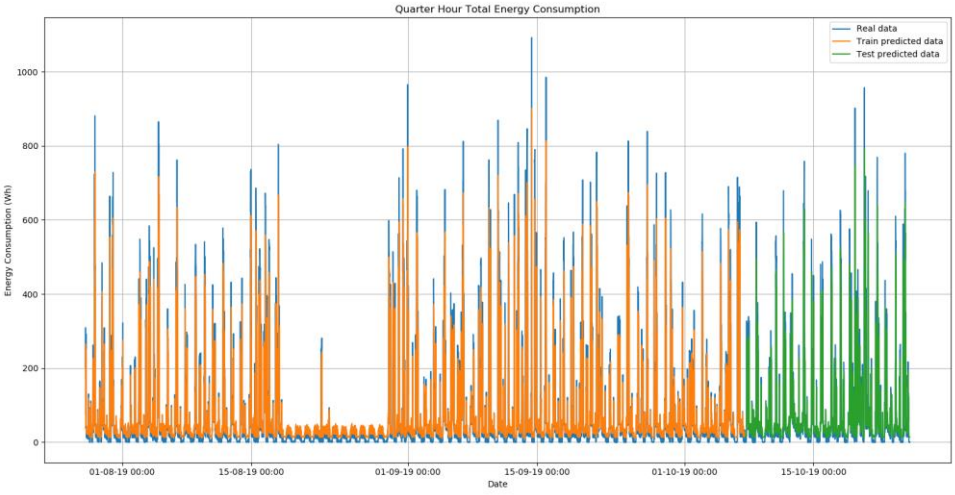

Figura 4.37 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *Linear Regression*.

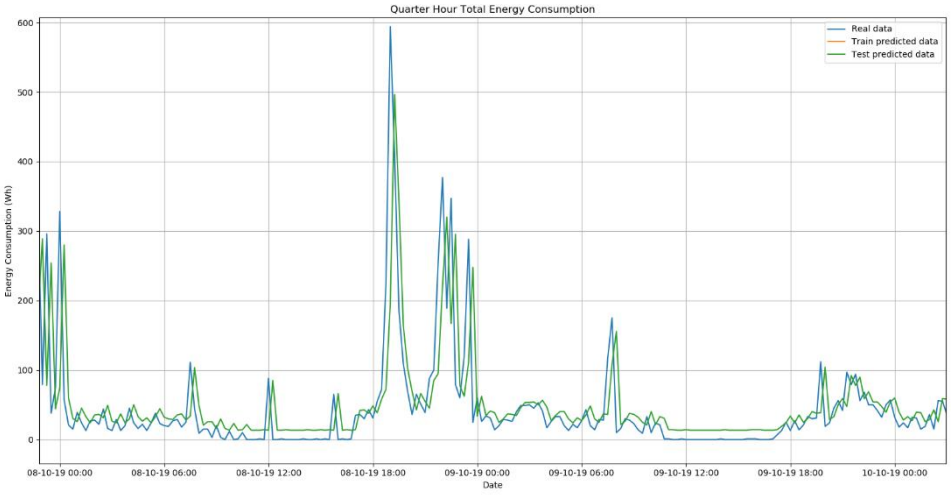

Figura 4.38 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *Linear Regression*.

Quanto ao *Facebook Prophet*, pela observação unicamente do gráfico da Figura 4.39, não é possível fazer comparações entre a previsão e os dados reais. Por outro lado, na Figura 4.40, verifica-se que o gráfico evidencia uma previsão com amplitude significativamente inferior aos dados reais. É de referir, ainda, que o padrão de previsão não acompanha os valores originais. Outro aspeto a referir, é o facto de os valores previstos não serem iguais a zero quando os originais o são.

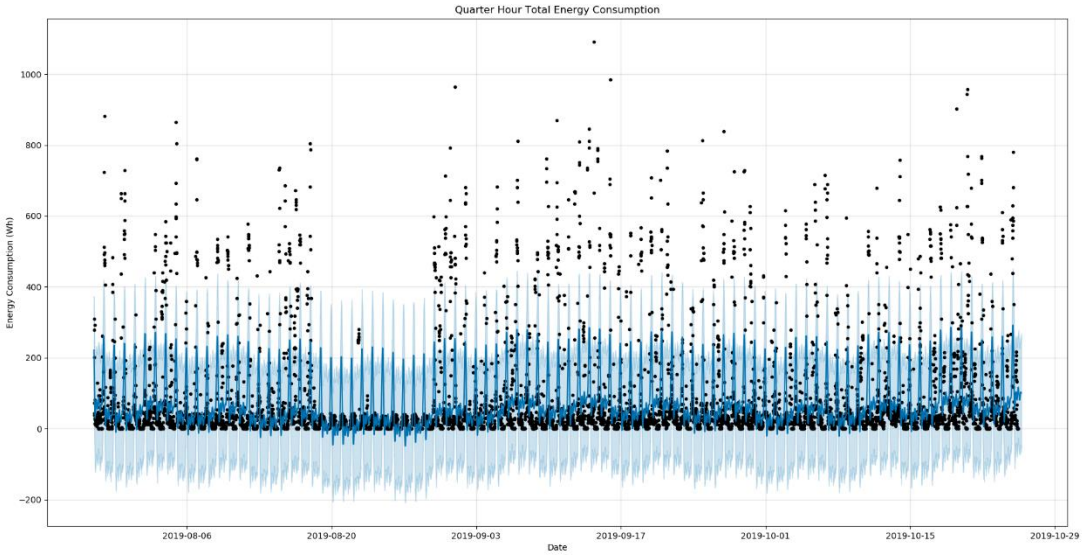

Figura 4.39 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *Facebook Prophet*.

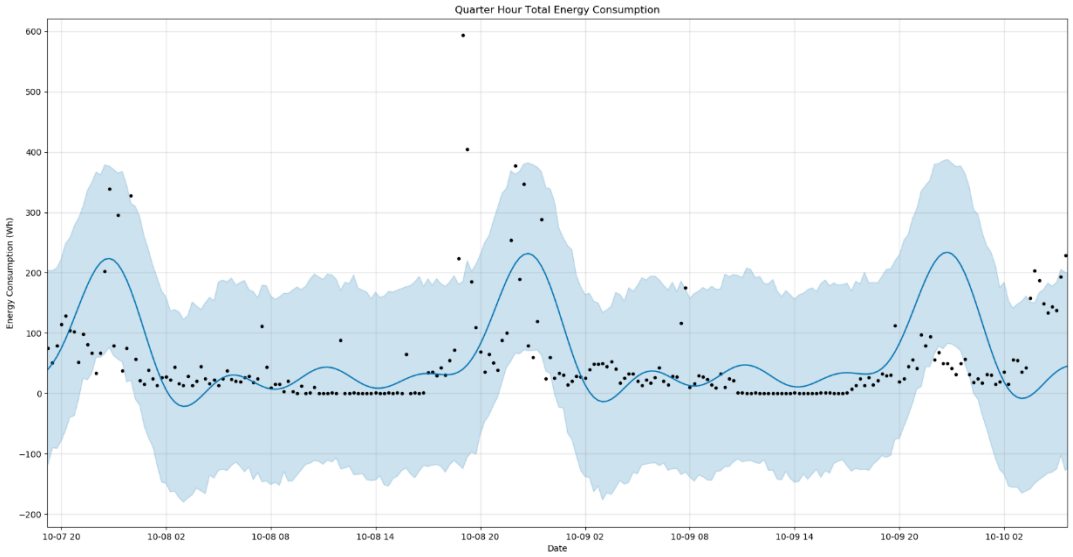

Figura 4.40 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *Facebook Prophet*.

No que respeita ao *SVR*, pela análise da Figura 4.41, identificam-se diferenças relativamente aos dados originais, em termos de amplitude, tal como na Figura 4.42. A razão deve-se ao facto de os valores previstos pelo algoritmo não alcançarem os picos de consumo e os valores mínimos se situarem acima dos dados históricos originais, sendo, por isso, uma solução que não responde ao problema.

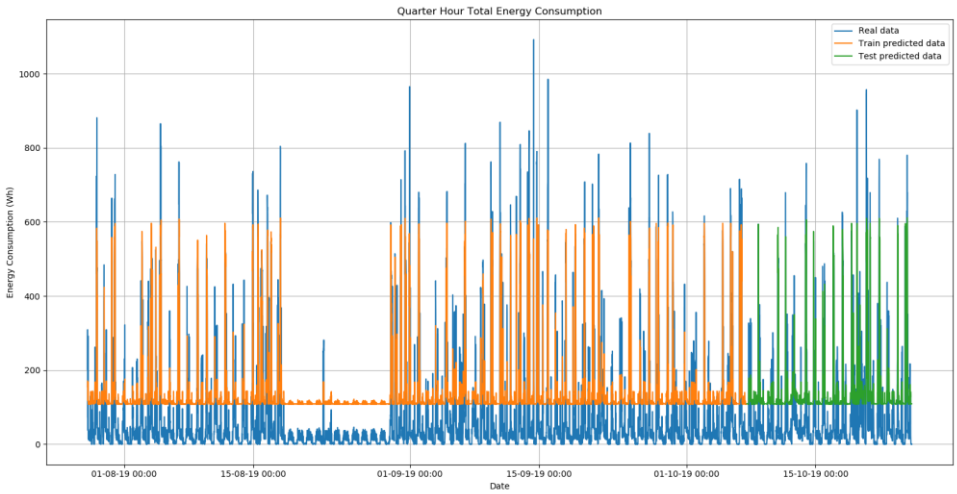

Figura 4.41 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *SVR*.

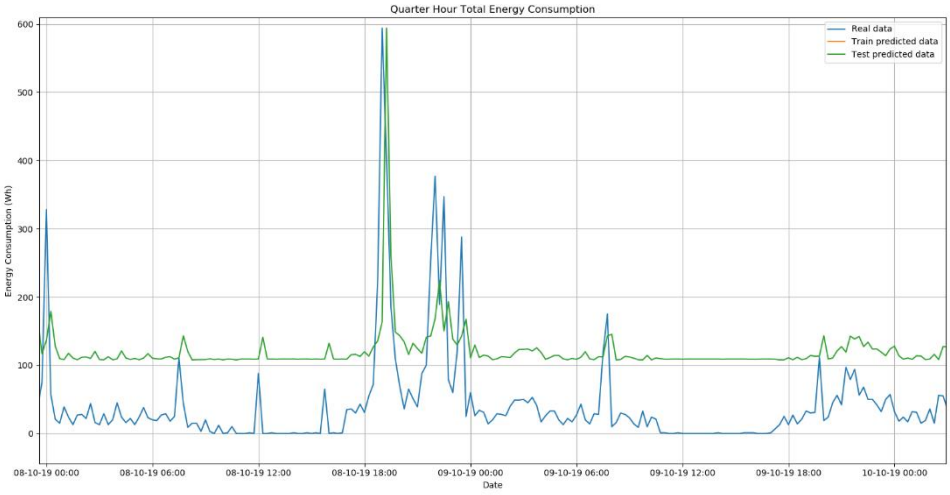

Figura 4.42 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *SVR*.

Quanto ao *Decision Tree*, a Figura 4.43 apresenta uma previsão semelhante ao *Random Forest*, no sentido em que a amplitude é ligeiramente inferior comparada com a dos dados fornecidos, incidindo maioritariamente nos valores máximos de consumo. Da mesma forma, a Figura 4.44 comprova essa semelhança, a qual demonstra a existência de um padrão semelhante ao da série original fornecida. Deste modo, considera-se que este algoritmo responde, favoravelmente, à exigência do problema.

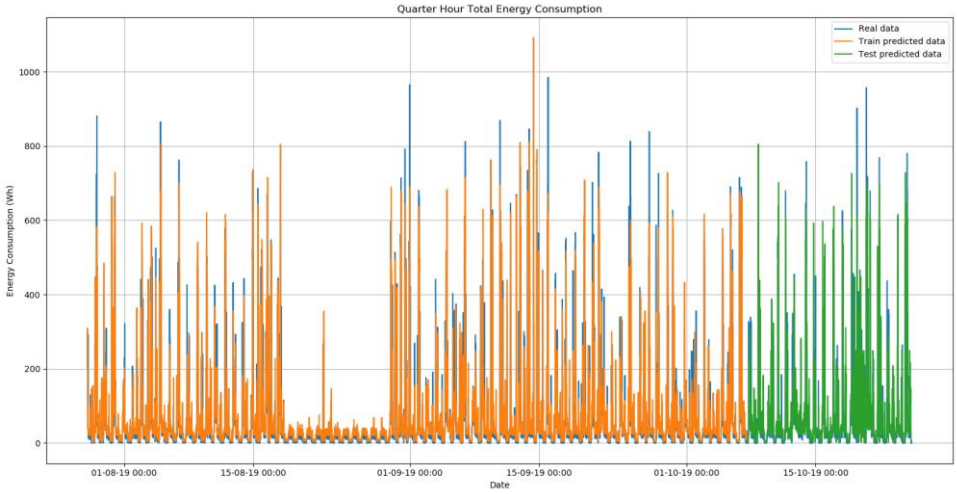

Figura 4.43 – Consumo real e previsão de energia durante três meses, com dados contínuos, modelo *Decision Tree*.

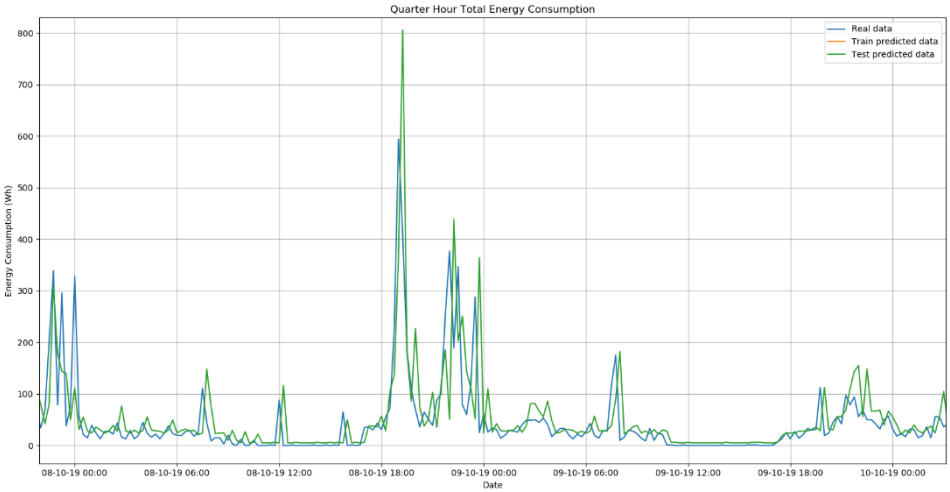

Figura 4.44 – Consumo real e previsão de energia durante dois dias consecutivos, com dados contínuos, modelo *Decision Tree*.

Esta série temporal tem uma duração de apenas três meses de registos, facto que não é muito recomendado, pois poderá resultar em previsões com menos precisão.

A Tabela 4.2 apresenta a síntese dos valores das métricas de erro calculadas para cada algoritmo.

Tabela 4.2 – Síntese dos resultados obtidos após a aplicação dos modelos utilizados para os dados contínuos.

Modelo	*RMSE*	*MSE*	*MAE*	*MAPE* (%)	R^2
***ANN* (Sem Ativação, *Adam*)**	83.25	6,930.90	43.77	198.71	0.65
***ANN* (Sem Ativação, *SGD*)**	127.29	16,203.86	76.98	488.02	0.18
***ANN* (*ReLU*, *Adam*)**	83.49	6,970.45	40.81	195.74	0.65
***ANN* (*ReLU*, *SGD*)**	127.95	16,369.93	76.42	476.15	0.18
***ANN* (*Sigmoid*, *Adam*)**	131.66	17,334.69	100.73	828.82	0.13
***ANN* (*Sigmoid*, *SGD*)**	202.92	41,175.83	191.64	1,855.95	-1.08
SVR	108.65	11,805.66	92.28	854.89	0.41
Decision Tree	95.95	9,206.17	46.78	120.86	0.54
Linear Regression	83.16	6,915.38	41.10	150.48	0.65
Random Forest	90.67	8,221.06	44.64	120.04	0.59
Facebook Prophet	120.51	14522.76	73.18	∞	0.25

Com a análise deste *data-set*, verifica-se um comportamento de consumo de energia diferente dos dados periódicos. Esta situação apresenta uma maior dificuldade nos treinos dos modelos, porque cada modelo pode introduzir ainda mais ruído e não contribuir para encontrar a melhor previsão.

Observa-se que existe uma certa consistência entre as diversas configurações de *ANN*, quer para o optimizador *Adam*, quer para *SGD*.

Constata-se, ainda, que o *Facebook Prophet* obteve um valor de *MAPE* infinito, visto que muitos valores originais fornecidos são iguais a zero.

Verifica-se que os melhores algoritmos são o *Decision Tree*, *Random Forest* e *Linear Regression*, com valores de *MAPE* (120.86%, 120.04% e 150.48%) e *RMSE* (95.95, 90.67 e 83.16), respetivamente. É de realçar que apresentam, também, resultados bastante favoráveis.

4.3 Dados descontínuos

4.3.1 Sem limitação do intervalo de tempo

Neste subcapítulo, nos dados provenientes dos equipamentos de medição de consumo energético, ao contrário dos anteriores, existem intervalos de tempo nas amostras, onde não existem registos de valores de energia. Quanto maior for o período sem registos, menor será a precisão na previsão dos modelos.

As figuras com numeração ímpar apresentam um intervalo de tempo entre o dia 8 de janeiro e o dia 6 de abril de 2020. Neste período existem três registos de energia com valores muito elevados relativamente aos restantes, o que dificulta a sua visualização em alguns gráficos, não sendo possível, por isso, retirar ilações concretas sobre o padrão de consumo e de previsão, devido ao ajustamento da escala.

Para os gráficos das figuras com numeração par, considerou-se um intervalo de tempo de três dias (2, 3 e 4 de fevereiro de 2020), em que existe um período sem registos de consumo real, pertencente ao conjunto de treino. Pretende-se

observar o comportamento de previsão do algoritmo nos períodos em que não existem registos.

No caso do *ANN*, sem função de ativação e com optimizador *Adam*, observa-se que, na Figura 4.45 e na Figura 4.46, os valores mínimos de previsão são idênticos aos fornecidos. Verifica-se, ainda, que os valores previstos se situam ligeiramente abaixo dos valores originais e que os valores máximos da previsão se situam abaixo dos valores máximos dos dados fornecidos. Salienta-se, ainda, que a previsão é nula, nos períodos em que não existem registos de consumo real.

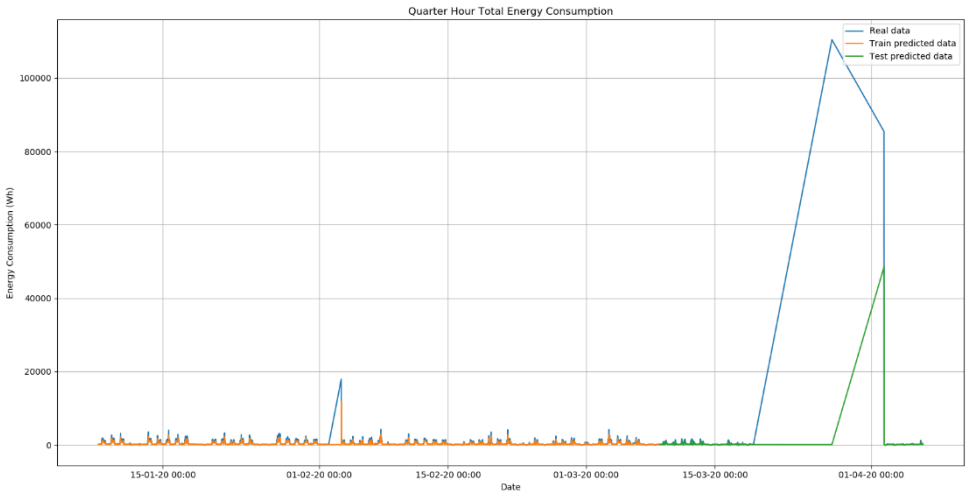

Figura 4.45 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *ANN*, optimizador *Adam*.

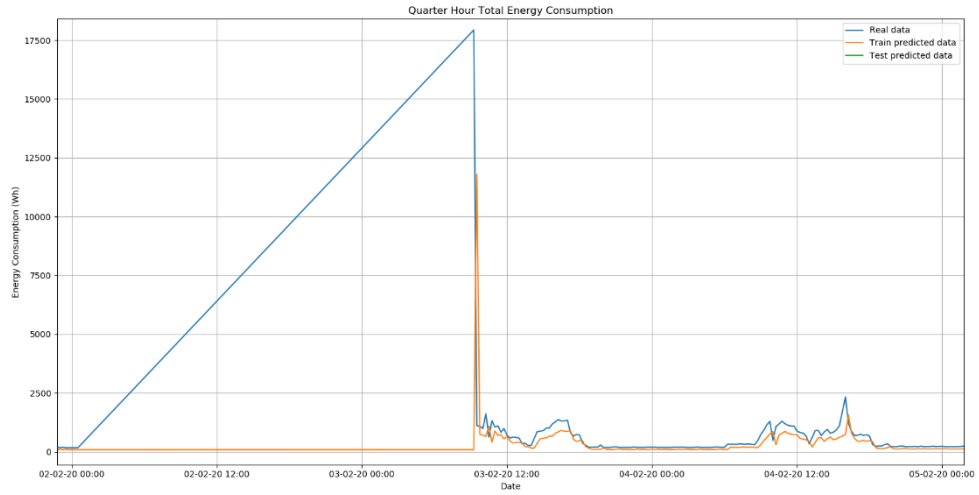

Figura 4.46 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *ANN*, optimizador *Adam*.

Relativamente ao *ANN,* com ativação *ReLU* e optimizador *Adam*, observa-se, na Figura 4.47 e na Figura 4.48, que os valores mínimos de previsão são similares aos dados reais. Verifica-se que os valores máximos da previsão se situam abaixo dos valores máximos dos dados fornecidos. À semelhança da configuração anterior, a previsão é nula nos intervalos de tempo em que não há registos de consumo real.

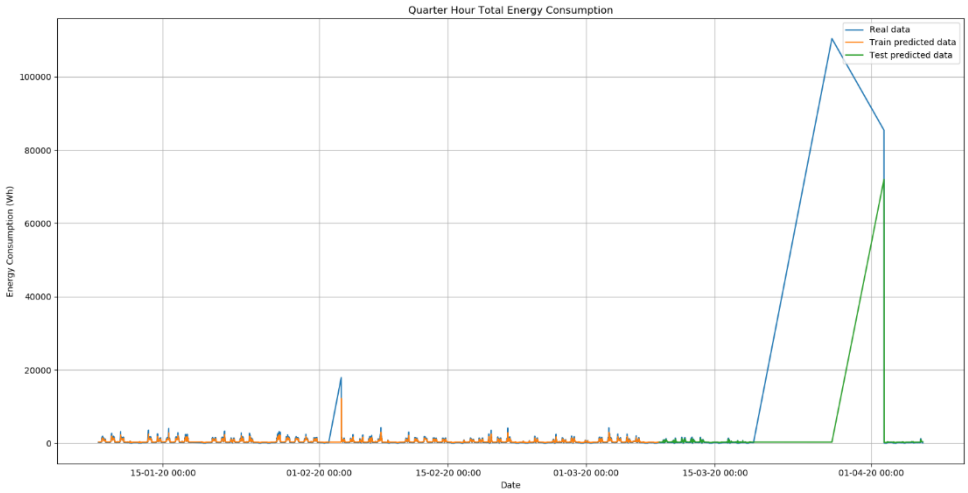

Figura 4.47 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

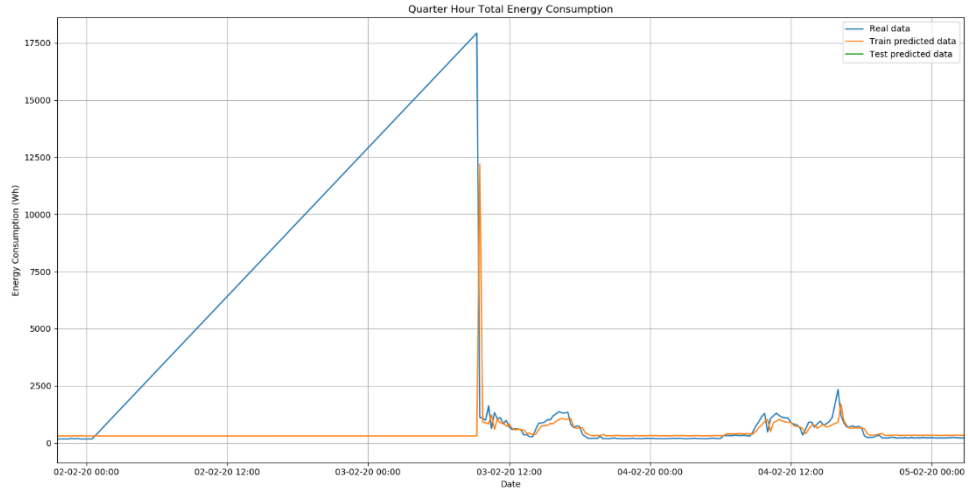

Figura 4.48 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

No que concerne ao *ANN*, com optimizador *SGD* e função de ativação *ReLU*, observa-se, na Figura 4.49, que os valores previstos são praticamente constantes ao longo do tempo. O mesmo se consegue observar, com mais detalhe, na Figura 4.50, na qual a previsão apresenta valores próximos de zero. Deste modo, conclui-se que esta configuração não responde ao problema.

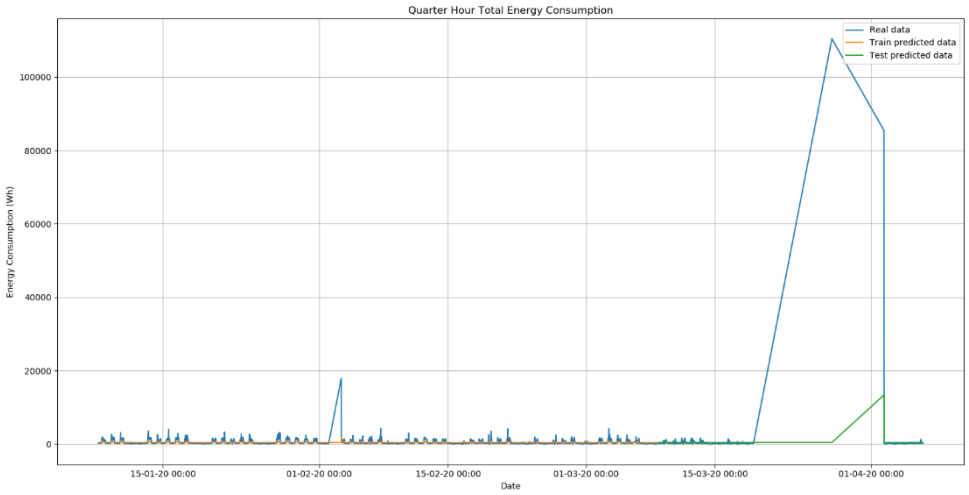

Figura 4.49 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

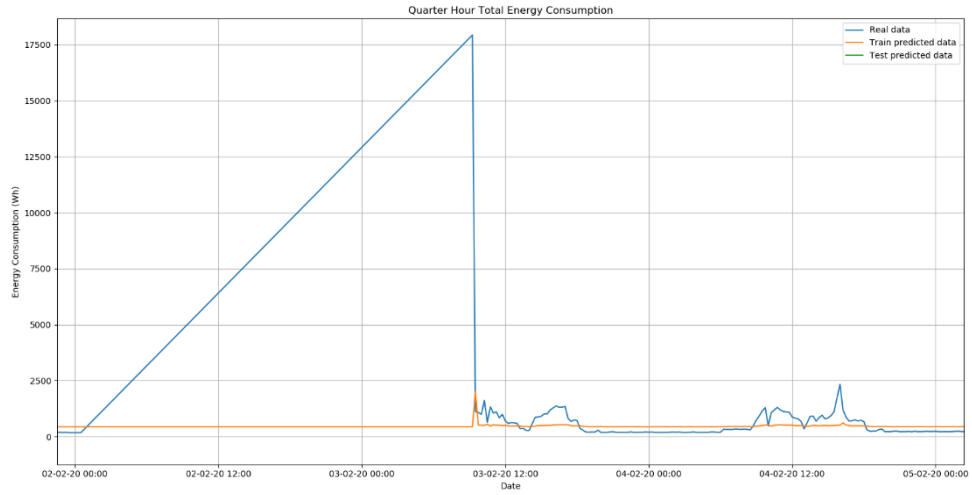

Figura 4.50 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

Em relação ao *ANN* com optimizador *SGD*, observa-se, na Figura 4.51, que os valores previstos permanecem quase constantes durante o intervalo de tempo considerado. Na Figura 4.52 verifica-se que a previsão demonstra valores quase nulos. Por conseguinte, esta configuração do algoritmo não é a adequada para solucionar o problema.

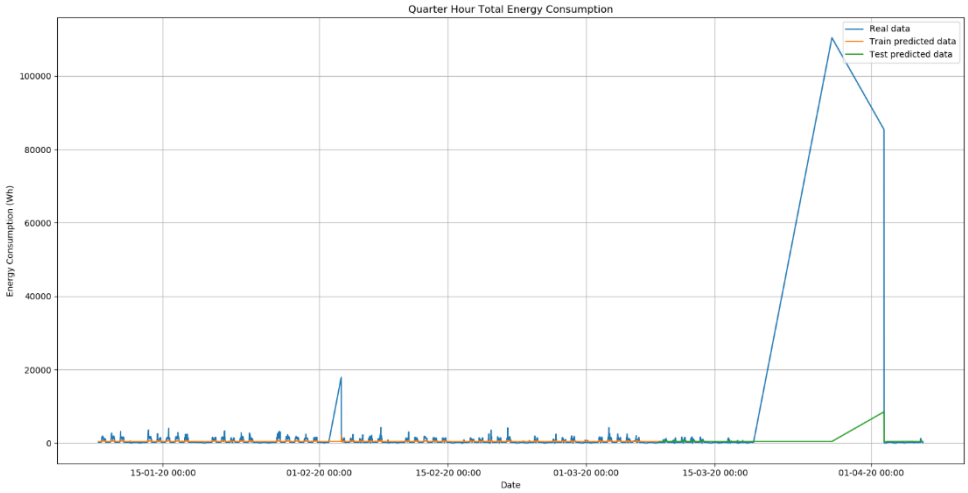

Figura 4.51 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *ANN*, optimizador *SGD*.

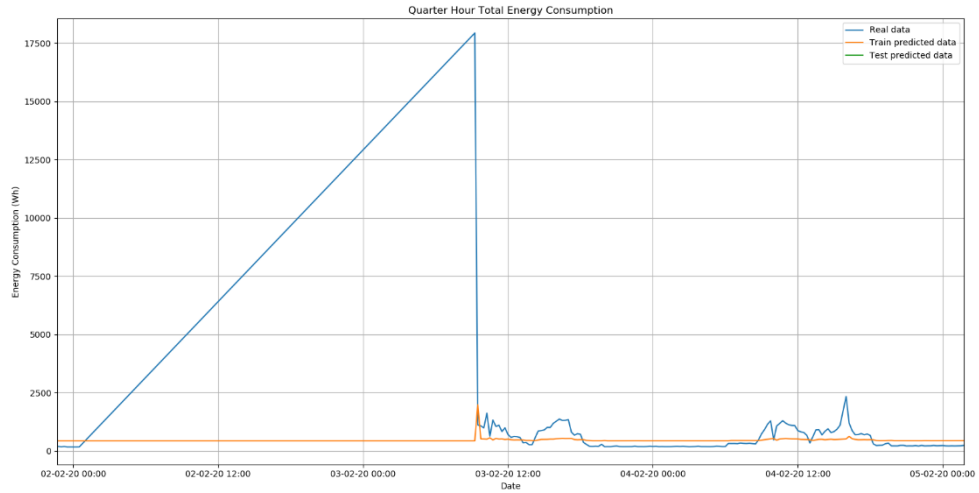

Figura 4.52 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *ANN*, optimizador *SGD*.

Relativamente ao *ANN*, com função de ativação *Sigmoid* e optimizador *Adam*, verifica-se, com base na Figura 4.53 e na Figura 4.54, que os valores da previsão se situam acima dos fornecidos e que o respetivo padrão se encontra invertido em relação ao padrão dos dados reais. Deste modo, esta configuração não soluciona o problema.

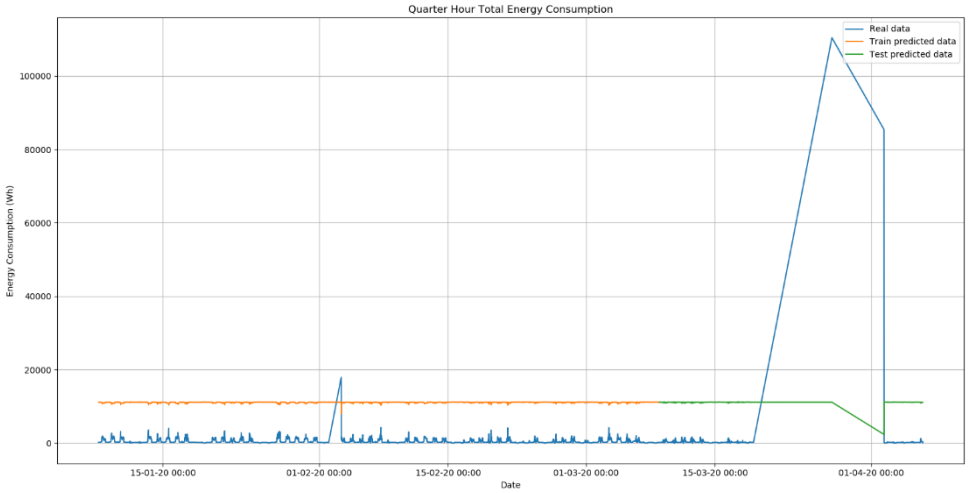

Figura 4.53 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

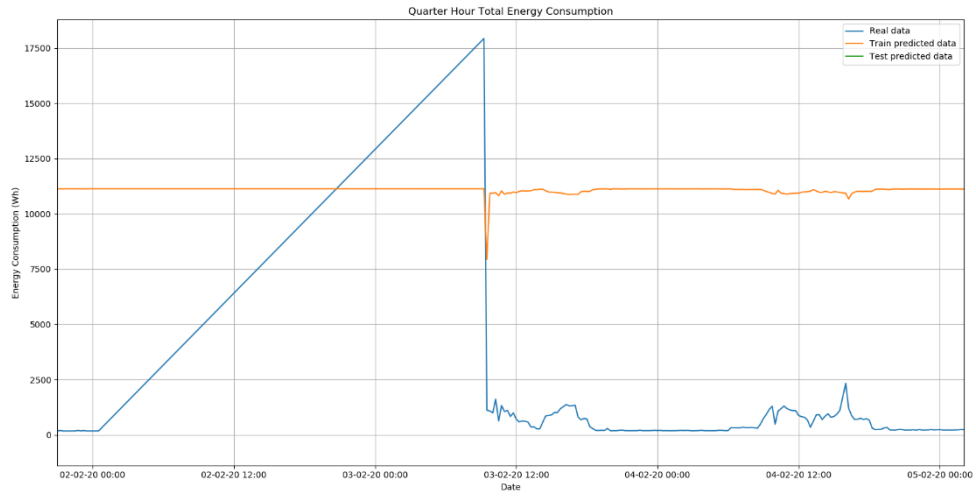

Figura 4.54 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

Relativamente à configuração de *ANN*, com função de ativação *Sigmoid* e optimizador *SGD*, verifica-se que, na Figura 4.55, os valores da previsão são constantes durante todo o intervalo de tempo. Da mesma forma, na Figura 4.56, é evidenciada a mesma situação, cujo valor de energia prevista se situa num patamar muito elevado. Assim, esta configuração não é desejável.

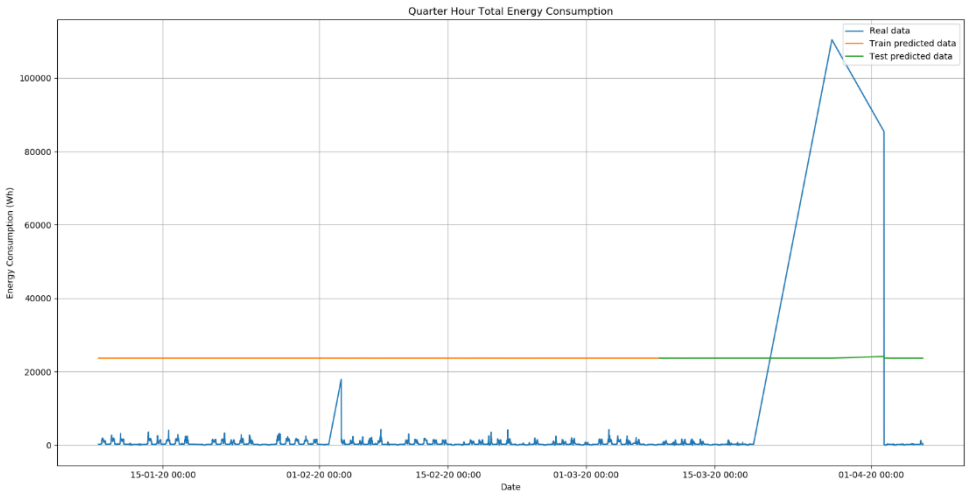

Figura 4.55 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

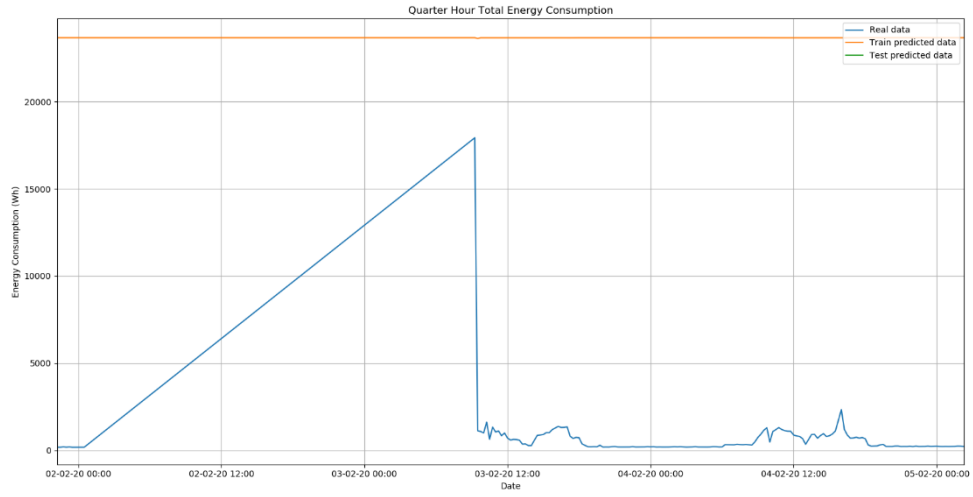

Figura 4.56 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

No que respeita ao *Random Forest*, observa-se que, na Figura 4.57 e na Figura 4.58, os valores mínimos da previsão se situam muito próximos dos valores mínimos originais. Verifica-se que a previsão se mantém praticamente constante e próxima de zero durante o intervalo de tempo em que não existem registos de consumo. Quando há registos de dados, a previsão acompanha o padrão do perfil de consumo.

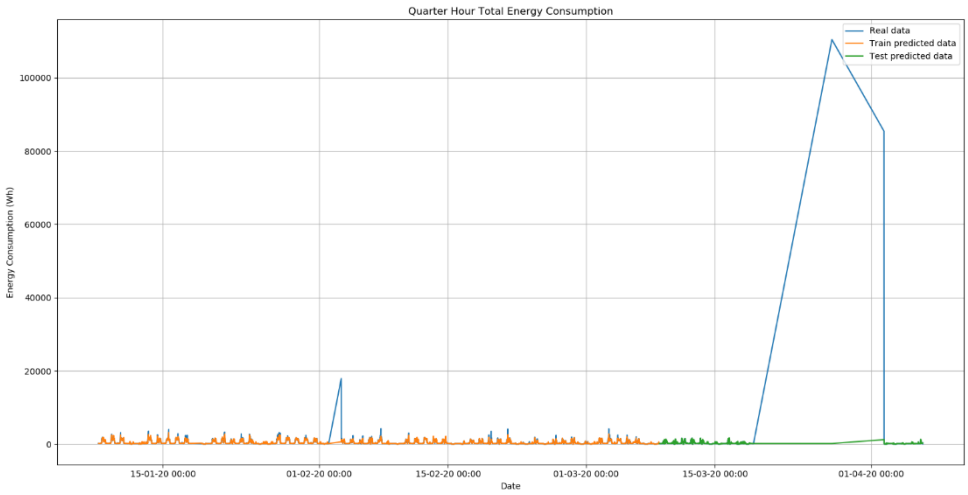

Figura 4.57 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *Random Forest*.

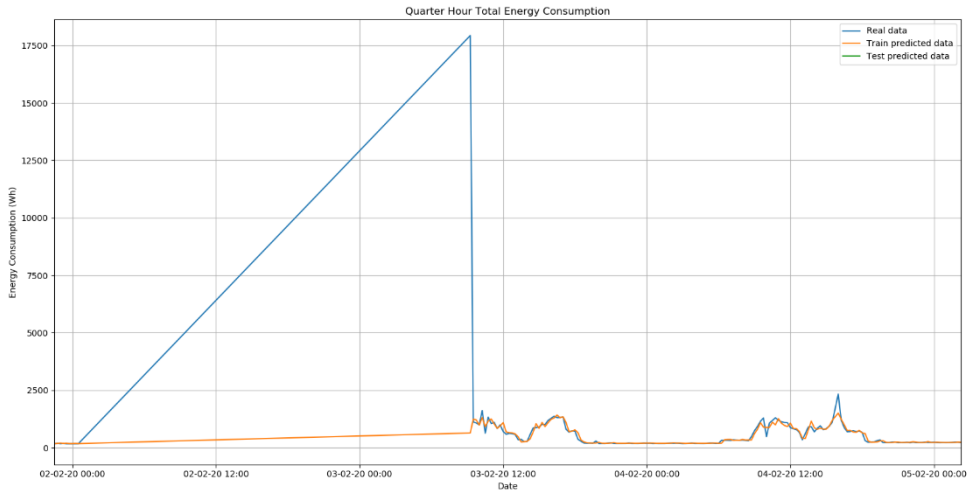

Figura 4.58 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *Random Forest*.

No caso do *Linear Regression*, na Figura 4.59 e na Figura 4.60, a previsão acompanha o padrão de consumo de registos fornecidos, mesmo nas situações em que existem picos de consumo real. A previsão mantém-se constante, com valores próximos de zero, durante o intervalo de tempo sem registos de consumo.

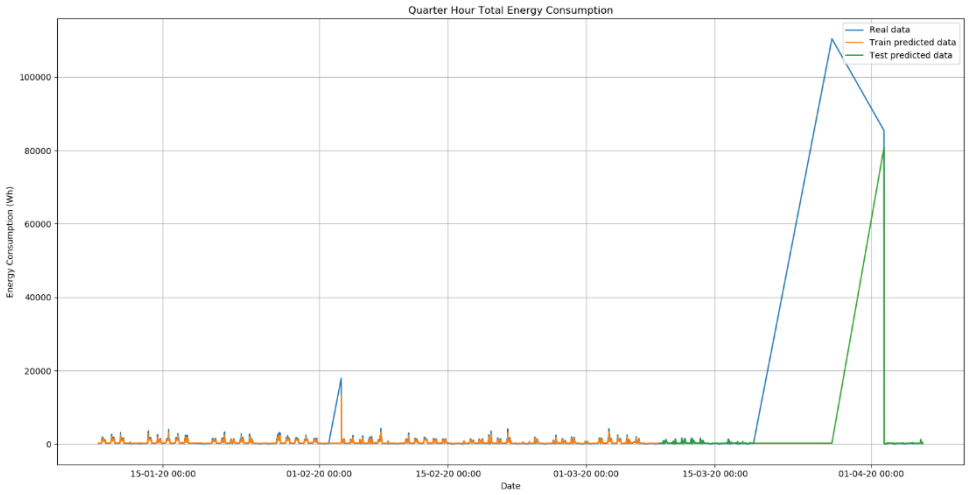

Figura 4.59 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *Linear Regression*.

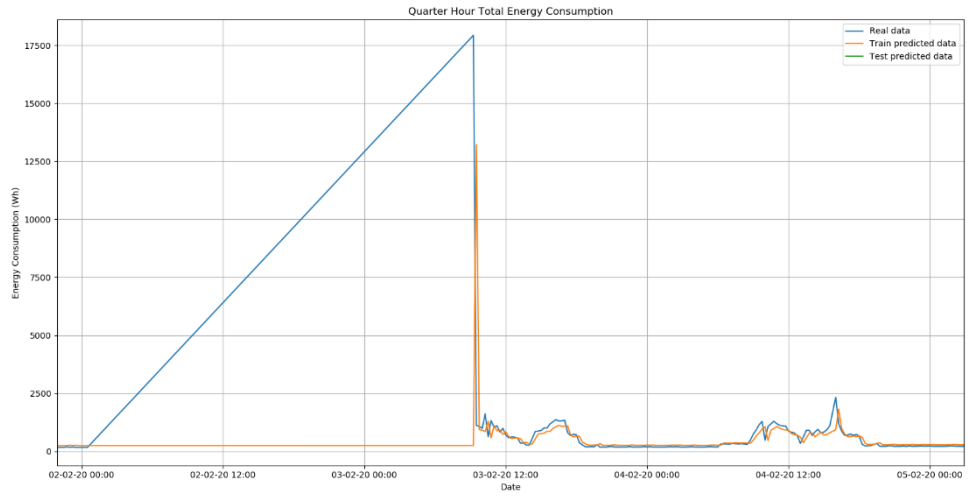

Figura 4.60 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *Linear Regression*.

No que respeita ao *Facebook Prophet*, verifica-se, com base na Figura 4.61 e na Figura 4.62, que a previsão acompanha o padrão de perfil de consumo, exceto nas situações isoladas de pico. A previsão mantém-se praticamente constante no intervalo de tempo sem registos, apresentando valores próximos de zero.

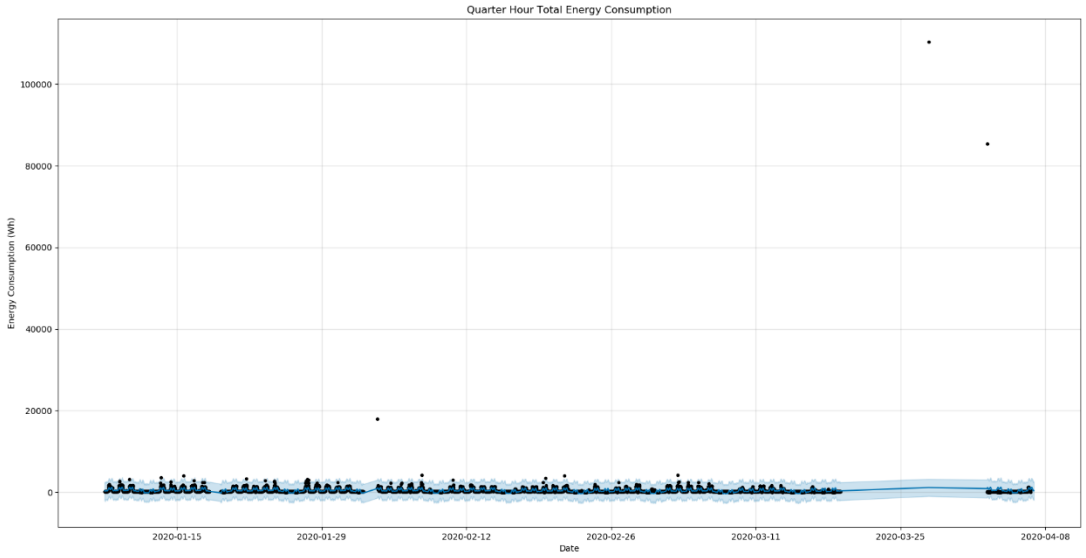

Figura 4.61 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *Facebook Prophet*.

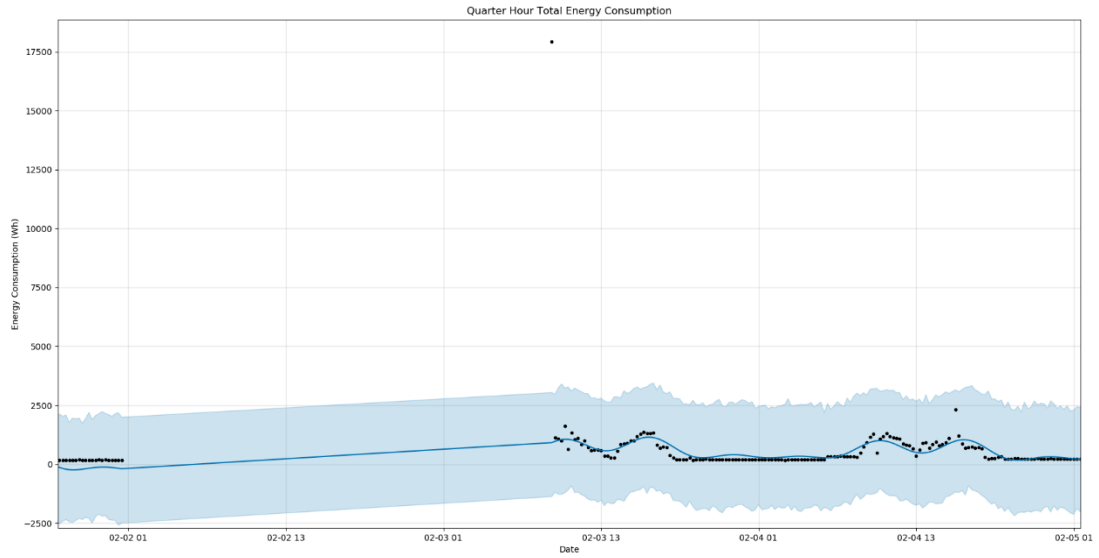

Figura 4.62 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *Facebook Prophet*.

Quanto ao *SVR*, ao observar a Figura 4.63, identifica-se uma previsão constante muito acima dos valores fornecidos, sendo mais visível ainda na Figura 4.64. Como tal, este algoritmo não é adequado para este problema.

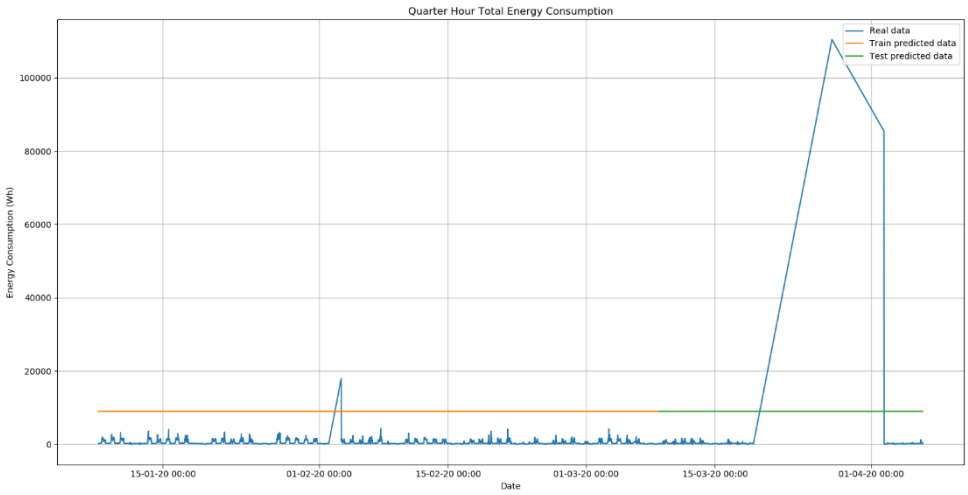

Figura 4.63 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *SVR*.

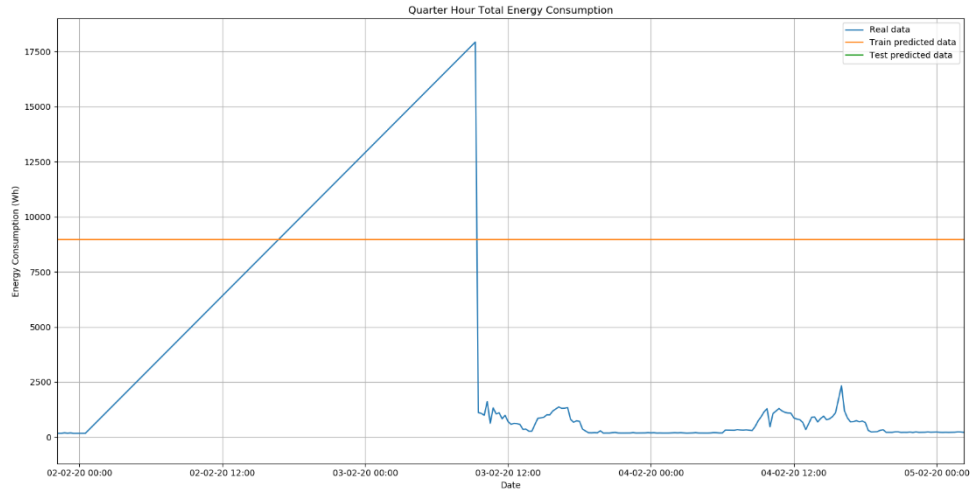

Figura 4.64 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *SVR*.

Em relação ao *Decision Tree*, na Figura 4.65 e na Figura 4.66, observa-se uma previsão similar ao padrão de dados de consumo fornecidos. Verifica-se que a previsão se mantém praticamente constante durante o intervalo de tempo sem registos de consumo, com valores próximos de zero.

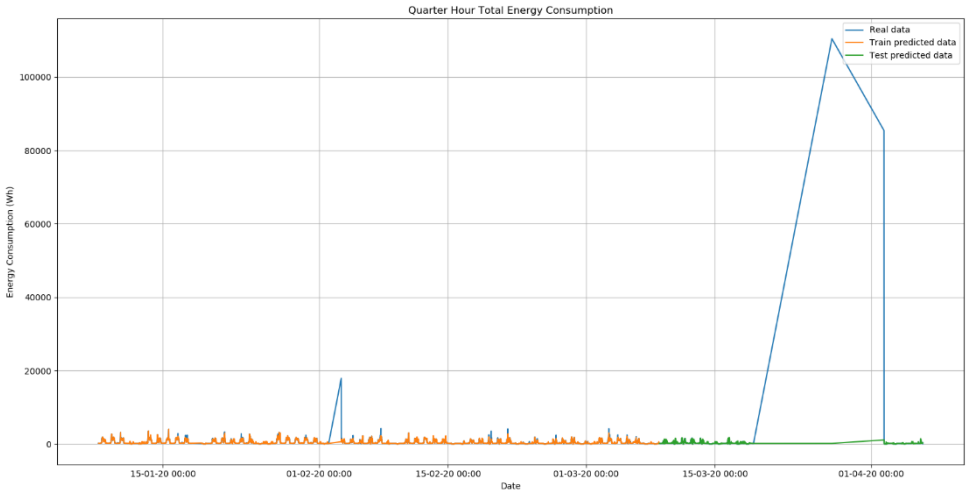

Figura 4.65 – Consumo real e previsão de energia durante três meses, com dados descontínuos, modelo *Decision Tree*.

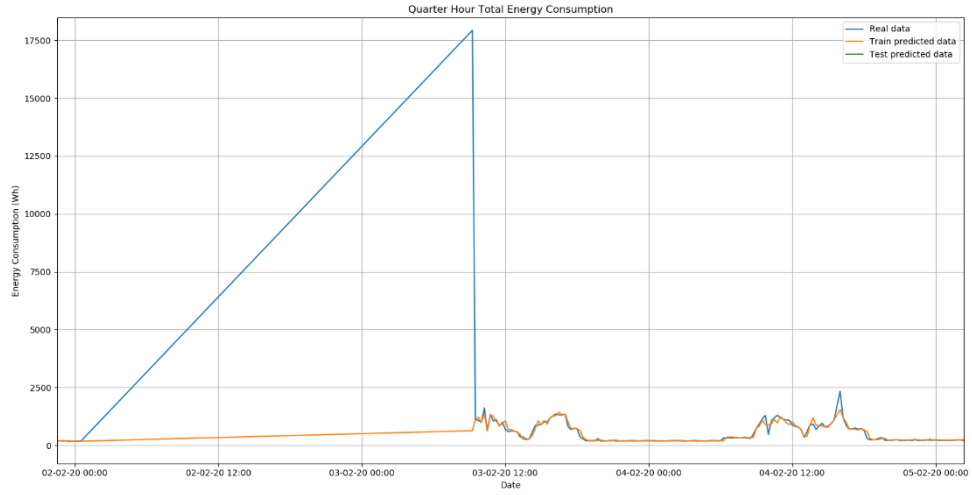

Figura 4.66 – Consumo real e previsão de energia durante três dias consecutivos, com dados descontínuos, modelo *Decision Tree*.

Assim, a Tabela 4.3 sintetiza os resultados obtidos das métricas de erro pela aplicação dos vários modelos utilizados, tendo como entrada os dados descontínuos.

Tabela 4.3 – Síntese dos resultados obtidos após a aplicação dos modelos utilizados para os dados descontínuos.

Modelo	*RMSE*	*MSE*	*MAE*	*MAPE* (%)	R^2
***ANN* (Sem Ativação, *Adam*)**	3,331.29	11,097,491.42	253.68	346.50	0.21
***ANN* (Sem Ativação, *SGD*)**	3,630.19	13,178,251.84	415.42	819.84	0.07
***ANN* (*ReLU*, *Adam*)**	3,370.58	11,360,794.72	277.43	429.12	0.19
***ANN* (*ReLU*, *SGD*)**	3,562.45	12,691,028.25	415.13	824.76	0.1
***ANN* (*Sigmoid*, *Adam*)**	11,397.62	129,905,723.18	10,969.31	23,810.59	-8.21
***ANN* (*Sigmoid*, *SGD*)**	23,493.4	551,940,061.5	23,408.3	50,464.29	-38.15
SVR	9,355.62	87,527,628	8,826.84	19,120.07	-5.21
Decision Tree	3,739.28	13,982,208.46	217.47	83.23	0.01
Linear Regression	3,411.71	11,639,738.41	232.19	300.99	0.17
Random Forest	3,736.77	13,963,423.72	213.68	84.35	0.01
Facebook Prophet	1,711.82	2,930,328.53	283.89	∞	0.04

Pelos resultados observados, podemos afirmar que nenhum algoritmo corresponde à resolução do problema. Verifica-se, todavia, que, mesmo obtendo resultados elevados, os melhores algoritmos são o *Linear Regression*, *Decision Tree* e *Random Forest*, com valores de *MAPE* (300.99%, 83.23% e 84.35%) e *RMSE* (3,411.71, 3,739.28 e 3,736.77), respetivamente.

Observa-se, ainda, que o *Facebook Prophet* obteve um valor de *MAPE* infinito, visto que muitos valores originais são equivalentes a zero.

4.3.2 Com limitação do intervalo de tempo

Os resultados obtidos a partir dos dados descontínuos, discutidos no ponto anterior, evidenciaram valores de erro muito elevados. Para minimizar o erro do problema, considerou-se somente o maior intervalo de tempo contínuo extraído dos dados descontínuos, sendo a duração de, aproximadamente, seis dias (12, 13, 14,

15, 16 e 17 de março de 2020). Como o conjunto de dados apresenta uma duração de seis dias, as sazonalidades semanais e mensais não podem ser testadas.

Em relação ao *ANN*, sem função de ativação e com optimizador *Adam*, observa-se que, na Figura 4.67, a previsão apresenta valores máximos muito abaixo dos valores originais e valores mínimos de previsão significativamente mais elevados que os reais, ou seja, a amplitude de previsão é bastante inferior à do consumo real. Assim, esta configuração não responde às necessidades do problema.

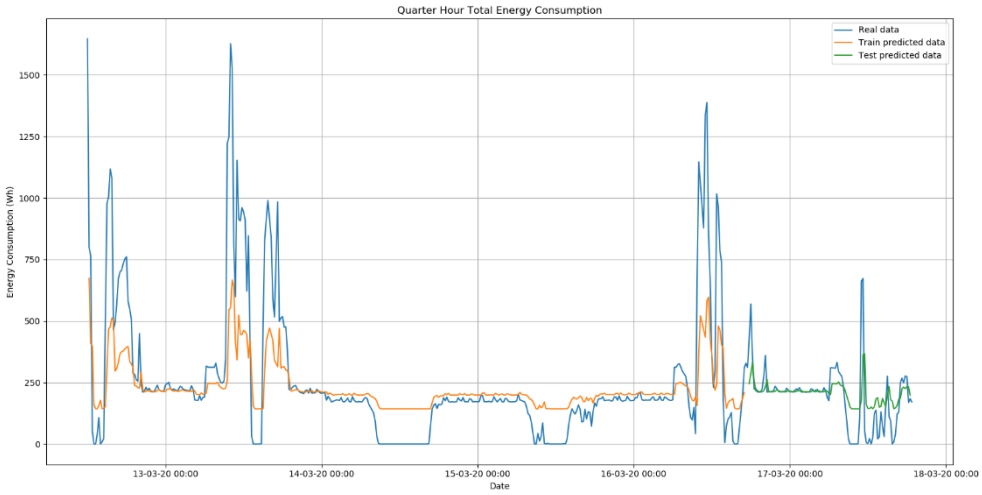

Figura 4.67 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *ANN*, optimizador *Adam*.

No que respeita ao *ANN,* mas, desta vez, com ativação *ReLU*, verifica-se que, na Figura 4.68, a previsão efetuada com esta configuração apresenta valores praticamente constantes durante o intervalo de tempo. Esta previsão não acompanha o padrão de perfil de consumo, o que torna esta configuração do algoritmo uma solução não-ideal para o problema.

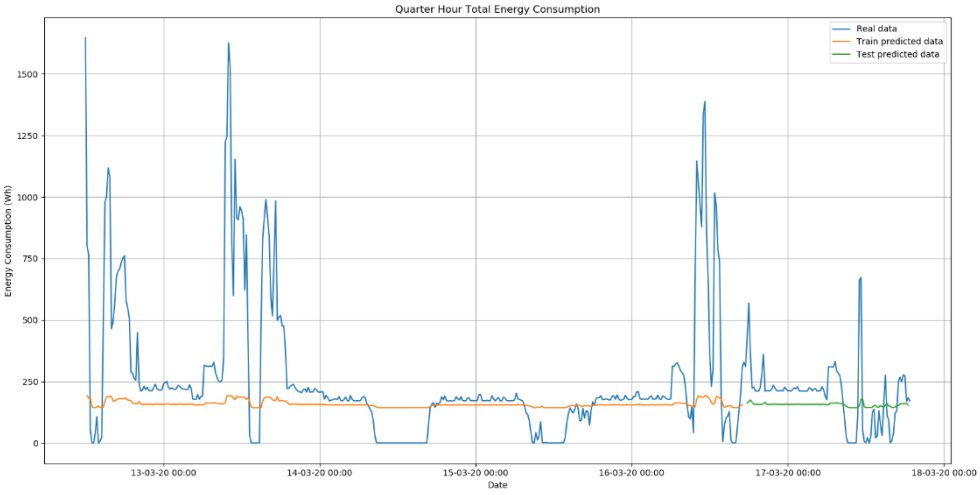

Figura 4.68 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *ANN*, ativação *ReLU* e optimizador *Adam*.

Relativamente ao *ANN*, com ativação *ReLU* e optimizador *SDG*, a Figura 4.69 apresenta valores de previsão de energia constantes durante o intervalo de tempo da amostra, não acompanhando o padrão de consumo da série temporal original. Assim, esta configuração do algoritmo não é recomendável.

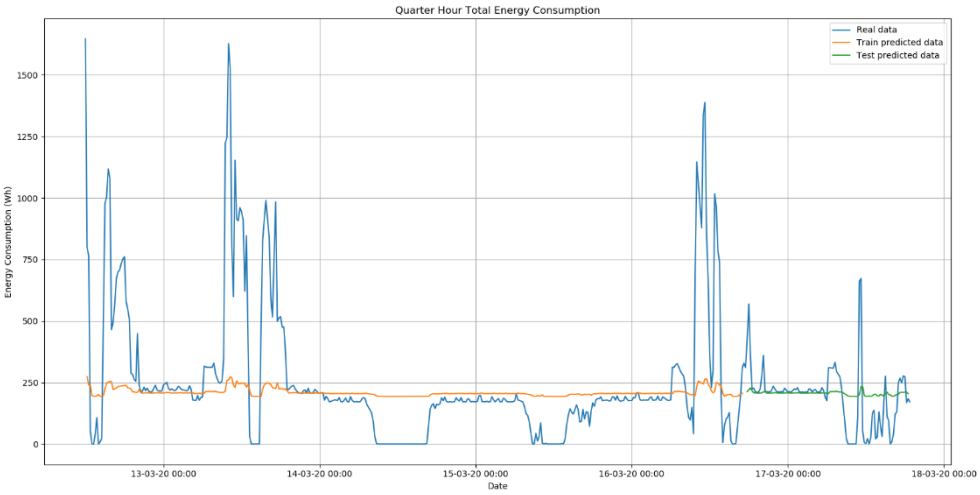

Figura 4.69 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *ANN*, ativação *ReLU* e optimizador *SGD*.

No que concerne ao *ANN*, com optimizador *SGD*, constata-se que, na Figura 4.70, a previsão não acompanha o padrão de perfil de consumo real, apresentando valores praticamente constantes durante o intervalo de tempo considerado. Esta configuração também não é solução para o problema.

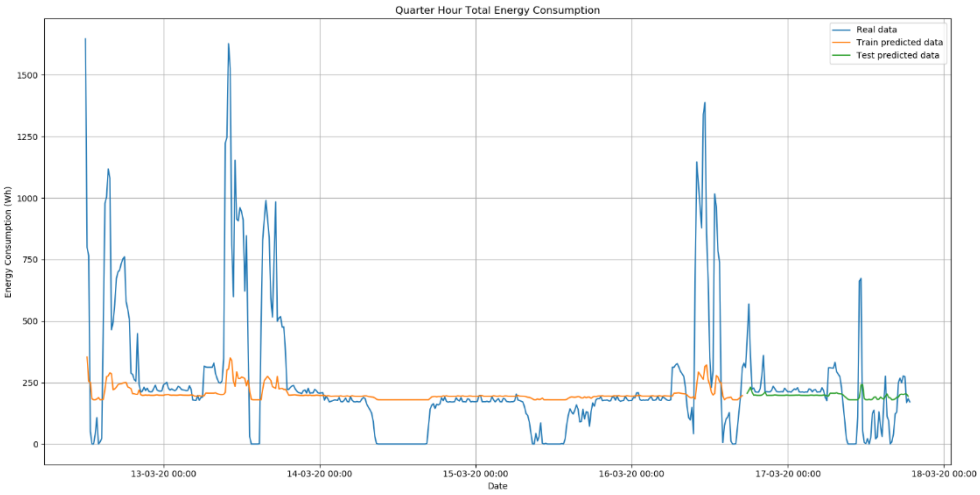

Figura 4.70 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *ANN*, optimizador *SGD*.

Relativamente ao *ANN*, com função de ativação *Sigmoid* e optimizador *Adam*, verifica-se que, na Figura 4.71, a previsão se mantém constante ao longo do intervalo de tempo, evidenciando valores elevados em relação aos dados reais, não acompanhando o padrão de consumo real. Assim, é de evitar esta configuração.

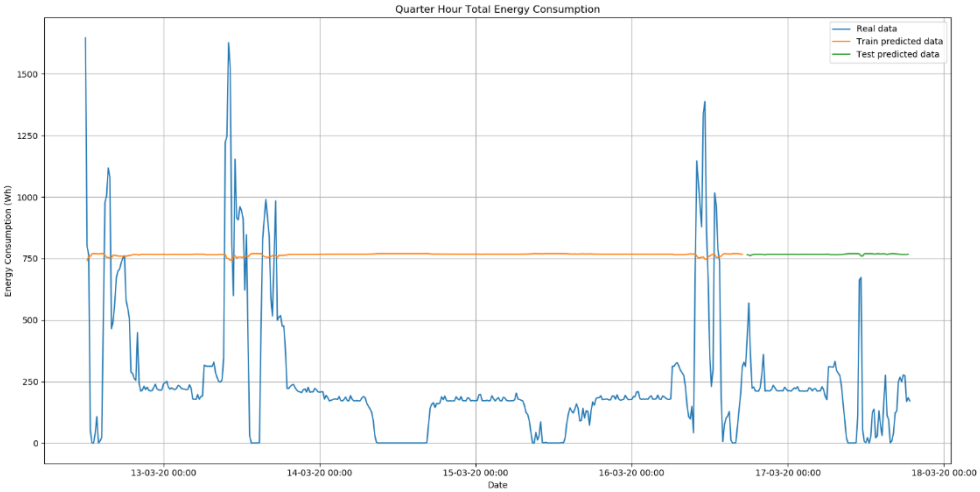

Figura 4.71 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *Adam*.

Quanto à utilização de *ANN*, com função de ativação *Sigmoid* e optimizador *SGD*, verifica-se que, na Figura 4.72, a previsão não acompanha o padrão de consumo real, mantendo-se constante durante o intervalo de tempo. Deste modo, esta configuração não pode ser tida em consideração para a solução do problema.

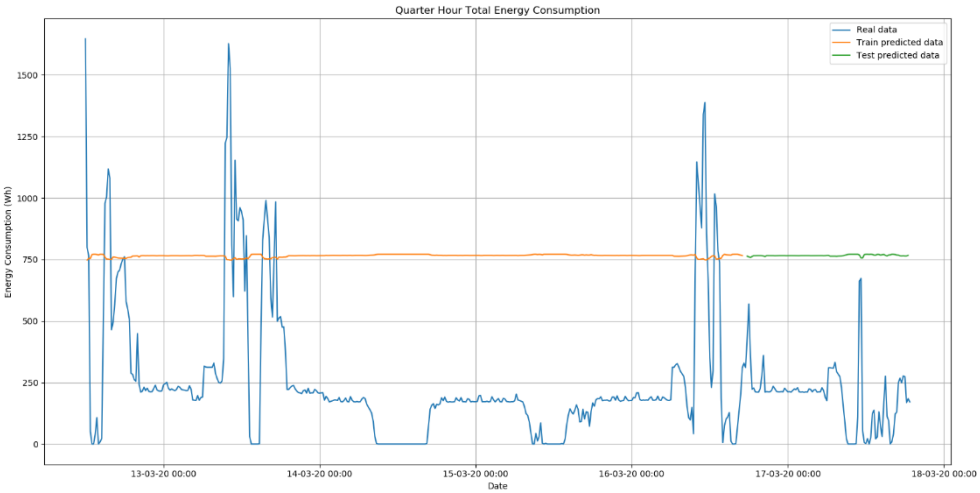

Figura 4.72 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *ANN*, ativação *Sigmoid* e optimizador *SGD*.

Em relação ao *Random Forest*, verifica-se que, na Figura 4.73, os valores da previsão se situam muito próximos dos originais, mantendo um padrão semelhante ao do consumo real. Assim, este algoritmo pode ser considerado para a resolução do problema.

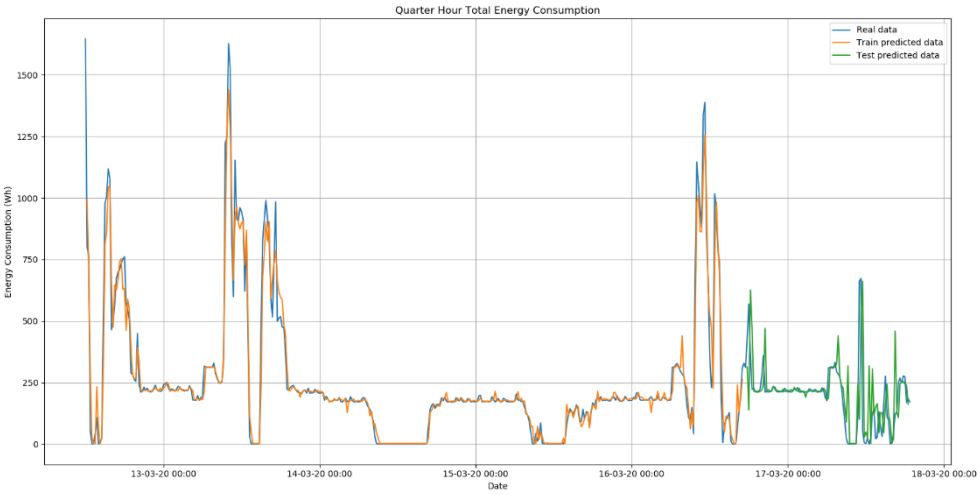

Figura 4.73 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *Random Forest*.

No caso do *Linear Regression*, a Figura 4.74 apresenta uma previsão muito próxima dos dados originais, acompanhando o padrão de perfil de consumo real. Desta forma, pode-se considerar este algoritmo para solucionar o problema.

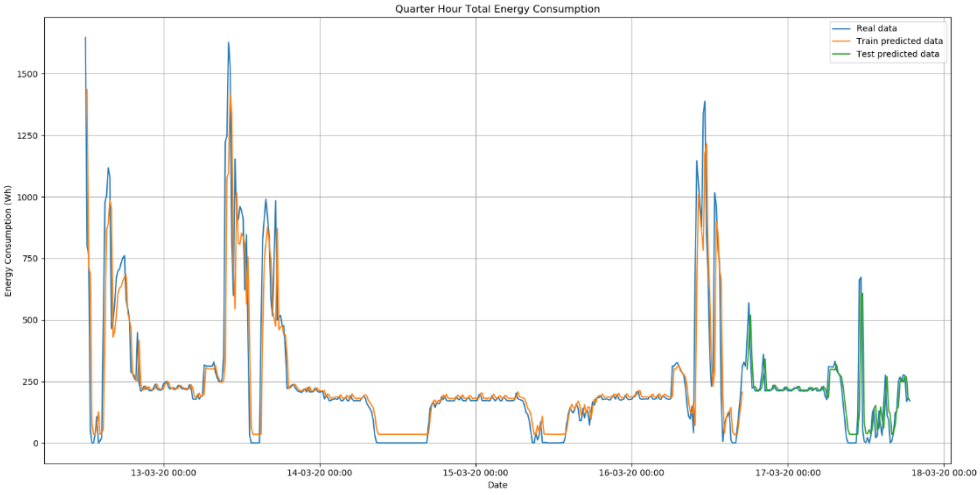

Figura 4.74 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *Linear Regression*.

Para o *Facebook Prophet*, observa-se, na Figura 4.75, que a amplitude da previsão é significativamente inferior à dos dados reais. É de referir, também, que o padrão de previsão não acompanha o padrão de consumo dos dados originais. Então, não é de considerar este algoritmo para a resolução do problema.

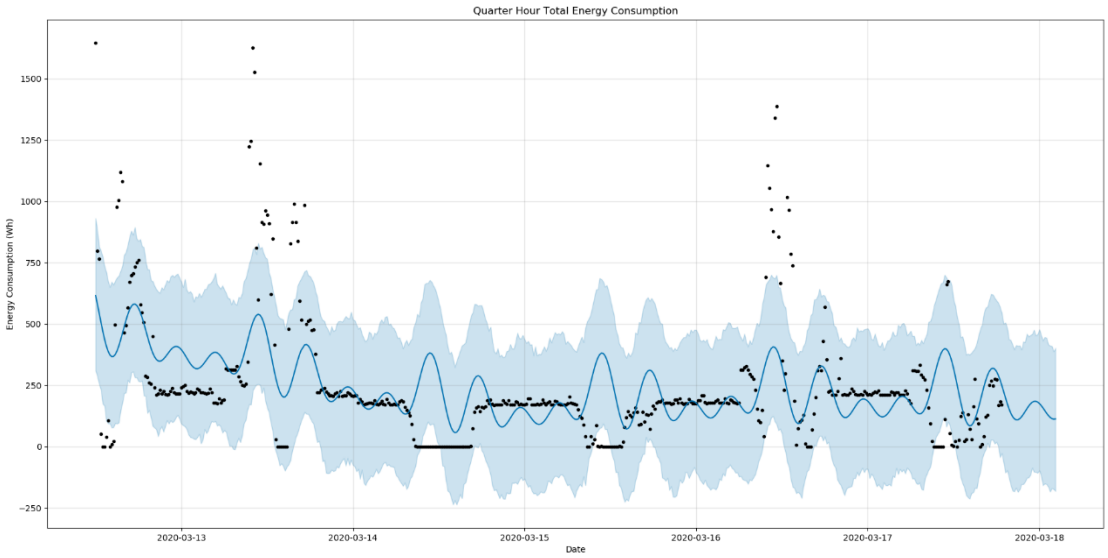

Figura 4.75 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *Facebook Prophet*.

No caso do *SVR*, ao observar a Figura 4.76, verifica-se que a previsão, a nível geral, acompanha o padrão de consumo real. No entanto, constata-se que esta previsão apresenta valores mínimos superiores aos registos mínimos reais, situação que não é a mais desejada para escolha do algoritmo.

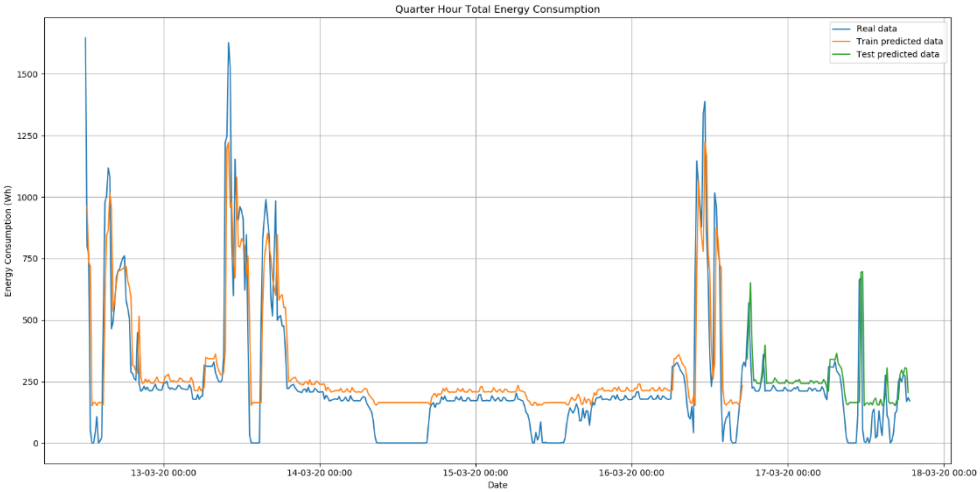

Figura 4.76 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *SVR*.

Relativamente ao *Decision Tree*, verifica-se que, na Figura 4.77, a previsão apresenta valores muito próximos dos fornecidos, ou seja, mantém um padrão semelhante ao do consumo real. Pode-se considerar que este algoritmo responde de forma satisfatória ao problema.

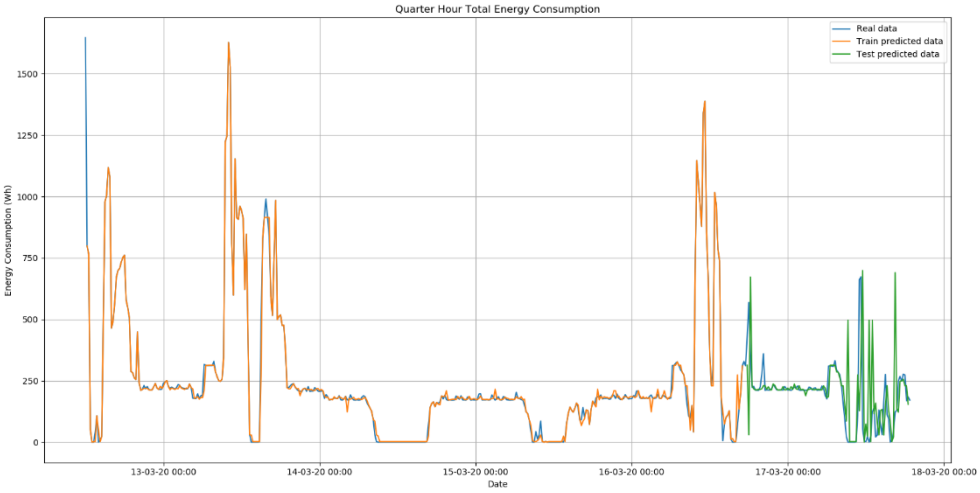

Figura 4.77 – Consumo real e previsão de energia durante seis dias, com dados contínuos provenientes dos dados descontínuos, modelo *Decision Tree*.

De referir que esta série temporal é constituída por um intervalo de tempo menor em relação às outras amostras de dados, originando previsões menos ajustadas ao padrão de perfil de dados de consumo real.

De seguida, apresentam-se, na Tabela 4.4, os resultados obtidos das métricas de erro para cada algoritmo.

Tabela 4.4 – Síntese dos resultados obtidos após a aplicação dos modelos utilizados para os dados descontínuos.

Modelo	*RMSE*	*MSE*	*MAE*	*MAPE* (%)	R^2
***ANN* (Sem Ativação, *Adam*)**	102.37	10,480.21	62.6	332.58	0.33
***ANN* (Sem Ativação, *SGD*)**	120.63	14,552.8	83.35	416.7	0.06
***ANN* (*ReLU*, *Adam*)**	125.34	15,711.26	96.35	325.58	-0.01
***ANN* (*ReLU*, *SGD*)**	120.25	14,460.90	78.51	438.04	0.07
***ANN* (*Sigmoid*, *Adam*)**	592.98	351,627.38	579.18	1,945.75	-21.64
***ANN* (*Sigmoid*, *SGD*)**	594.94	353,955.53	582.02	1,935.43	-21.79
SVR	119.31	14,234.04	77.18	379.57	0.08
Decision Tree	154.24	23,788.74	68.95	95.98	-0.53
Linear Regression	94.83	8,993.15	46.44	116.31	0.42
Random Forest	129.8	16,847.64	64.01	85.43	-0.08
Facebook Prophet	226.99	51,527.39	145.85	∞	0.25

Observando a Tabela 4.4, pode-se afirmar que todas as configurações relativas ao modelo *ANN* apresentam os piores resultados, tornando-se num algoritmo a não considerar para a previsão com intervalo de tempo de valores históricos mais reduzido.

Verifica-se, ainda, que o *Facebook Prophet*, obteve um valor de *MAPE* infinito, razão justificada pela abundância de valores equivalentes a zero nos dados originais.

Concluindo, constata-se que os algoritmos que apresentam melhores resultados são o *Linear Regression*, *Decision Tree* e *Random Forest*, com valores de *MAPE* (116.31%, 95.98% e 85.45%) e *RMSE* (94.83, 154.24 e 129.8), respetivamente.

4.4 Síntese da discussão de resultados

A partir dos resultados apresentados pela Tabela 4.1, Tabela 4.2, Tabela 4.3 e Tabela 4.4, elaborou-se a Tabela 4.5, que apresenta um resumo do desempenho de previsão dos três algoritmos com mais precisão para cada série temporal. A seleção destes modelos foi realizada tendo em conta as métricas de erro *RMSE*, *MAPE* e R^2.

Tabela 4.5 – Síntese dos resultados obtidos após a aplicação dos algoritmos sobre o conjunto de dados para teste para cada série temporal.

Série de dados	Modelo	*RMSE*	*MAPE* (%)	R^2
Dados periódicos	*Decision Tree*	0	0	1
	Random Forest	0	0	1
	Linear Regression	3.41	2.07	0.99
Dados contínuos	*Linear Regression*	83.16	150.48	0.65
	Random Forest	90.67	120.04	0.59
	Decision Tree	95.95	120.86	0.54
Dados descontínuos	*Linear Regression*	3,411.71	300.99	0.17
	Random Forest	3,736.77	84.35	0.01
	Decision Tree	3,739.28	83.23	0.01
Dados contínuos provenientes de dados descontínuos	*Linear Regression*	94.83	116.31	0.42
	Random Forest	129.8	85.43	-0.08
	Decision Tree	154.24	95.98	-0.53

Para as séries temporais periódicas e contínuas, as diferenças de erro entre os valores dos três melhores algoritmos não são significativas. Em relação às séries temporais descontínuas e contínuas provenientes de dados descontínuos, as diferenças de erro das métricas são mais visíveis, nomeadamente, para as métricas de erro *RMSE*, *MAPE* e R^2.

De referir, ainda, que a eficácia de todos os algoritmos utilizados varia consoante a duração do intervalo de tempo existente, na série temporal utilizada.

Os modelos que demonstram melhor desempenho são o *Decision Tree*, o *Random Forest* e o *Linear Regression*. Esta situação reforça a importância de cada um na realização de previsões, tanto para conjuntos de dados com um intervalo de tempo superior, neste caso de três meses, como para conjuntos de dados com um intervalo de tempo inferior, neste caso de seis dias, quer sejam contínuos ou descontínuos.

4.5 Síntese do capítulo

Neste capítulo, foram apresentados e discutidos os resultados obtidos, utilizando os algoritmos propostos no capítulo anterior. Foram, também, calculadas diversas métricas de erro, nomeadamente, *RMSE*, *MAE*, *MSE*, *MAPE* e R^2, para todos os algoritmos utilizados, de forma a poder concluir quais os que apresentam melhores resultados para as diferentes séries temporais.

Foi ainda efetuada uma síntese dos resultados obtidos, para cada série temporal, com o intuito de averiguar quais os algoritmos que apresentam previsões com mais precisão.

Por último, a resposta à questão inicial deste trabalho, os objetivos inicialmente definidos e as propostas para futuras investigações neste contexto, serão explicitadas no capítulo seguinte.

5 Conclusão

De acordo com o que se descreveu anteriormente, pretende-se que as conclusões deste trabalho constituam, não só um momento de síntese e articulação de informações, mas também de interpretação e de levantamento de questões sobre as mesmas.

Começam-se as conclusões com base no que foi referido até agora, destacando-se a importância do ato de projetar que, basicamente, consiste no desenvolvimento de um processo elaborado, criativo e exigente.

Neste trabalho, foi feito um estudo teórico sobre os diversos algoritmos de *Machine Learning*, as suas vantagens e desvantagens e a forma de avaliação e validação dos mesmos.

No seguimento da análise da literatura, foi possível escolher os algoritmos que são mais referidos na comunidade científica. Depois, definiu-se a metodologia a adotar na implementação dos algoritmos escolhidos, usando os dados de consumo energético fornecidos, bem como conjuntos de dados criados a partir dos mesmos.

Após a implementação dos modelos, prosseguiu-se à sua avaliação, utilizando métricas de erro já definidas, sendo estas o *RMSE*, *MSE*, *MAE*, *MAPE* e R^2.

Através da análise comparativa dos diferentes resultados das métricas de erro (*RMSE*, *MAPE* e R^2) para cada algoritmo, foi possível identificar aqueles que apresentaram previsões com menor valor de erro.

Finalmente, o algoritmo escolhido foi o *Linear Regression*, pois apresenta valores de *RMSE* mais baixos para as séries temporais de consumo real, sendo estas as séries contínuas, descontínuas e contínuas provenientes de descontínuas. Este algoritmo foi, então, utilizado para realizar previsões relativamente aos dados reais das diferentes fases de uma instalação trifásica, a fim de selecionar aquela que apresenta um maior consumo.

Este trabalho permitiu a criação de uma estratégia de previsão de energia de *Machine Learning* que é capaz de medir previsões com bastante precisão. De notar que a fiabilidade aumenta, quando se usam dados reais, não surgindo problemas no treino do algoritmo.

De forma a responder à questão colocada no início deste trabalho, é necessária a existência de um programa que consiga escolher entre as diferentes fases da instalação trifásica, tendo em conta os valores originais de cada fase e os valores previstos pelo algoritmo escolhido de *Machine Learning*.

Considera-se, ainda, que a conclusão não ficaria completa se não se relembrasse os objetivos de partida, os quais serviram de fio condutor na pesquisa para a confrontar com os resultados de estudo. O objetivo central proposto foi elaborar um sistema de comutação de fases de uma instalação trifásica, através de *Machine Learning*. Os objetivos específicos definidos foram os seguintes:

1. Adotar modelos de *Machine Learning*, de modo a encontrar os que apresentam a melhor previsão.
2. Utilizar diferentes amostras de dados (periódicas, contínuas, descontínuas e contínuas provenientes de descontínuas), de modo a observar diversas previsões para cada caso.
3. Criar sistemas de previsão que permitam realizar previsões com base nos modelos usados, bem como a seleção de fases de uma instalação trifásica, tendo em conta os valores de energia previstos para cada fase.
4. Ensaiar e validar as previsões nos seguintes ambientes:
    - Numa instalação elétrica real, equipada com um seletor de fases que executará as instruções produzidas pelo modelo.
    - Em ambiente simulado, utilizando o perfil de consumo de instalações trifásicas e comparando-o com as estimativas produzidas pelo modelo.

Assim, em relação ao primeiro objetivo específico que se definiu, foram implementados vários modelos de *Machine Learning*, nomeadamente, *SVM*, *ANN*, *Decision Tree*, *Random Forest*, *Linear Regression* e *Facebook Prophet*. Conclui-se que nem todos os algoritmos apresentaram a mesma precisão para diferentes séries temporais, sendo os algoritmos *Decision Tree*, *Random Forest* e *Linear Regression* aqueles que apresentaram melhores resultados.

Quanto ao segundo objetivo específico “Utilizar diferentes amostras de dados (periódicas, contínuas, descontínuas e contínuas provenientes de descontínuas), de modo a observar diversas previsões para cada caso.”, ao longo desta investigação, foram utilizados diferentes tipos de amostras de dados. Os dados contínuos e descontínuos foram fornecidos pela empresa *Withus*, os dados periódicos foram criados a partir dos dados contínuos, por um programa em *Python 3*, em separado, e a série temporal contínua proveniente de dados descontínuos foi criada ao selecionar somente o maior intervalo de tempo contínuo da série temporal descontínua.

Relativamente ao terceiro objetivo “Criar sistemas de previsão que permitam realizar previsões com base nos modelos usados, bem como a seleção de fases de uma instalação trifásica, tendo em conta os valores de energia previstos para cada fase.”, foram criados dois programas em *Python 3*, sendo que um utiliza os diversos algoritmos e diferentes métricas de erro (nomeadamente, *RMSE*, *MSE*, *MAE*, *MAPE* e R^2) referidas anteriormente neste projeto, a fim de encontrar o melhor modelo a utilizar para cada amostra de dados, e outro para escolher a fase com base nos algoritmos selecionados anteriormente.

No que concerne ao objetivo “Ensaiar e validar as previsões nos seguintes ambientes”, conseguiu-se ensaiar e validar as previsões em ambiente simulado, utilizando séries temporais fornecidas e criadas. No entanto, não se conseguiu realizar o mesmo processo de ensaio e validação em sistemas trifásicos, devido à obrigatoriedade de confinamento decretado pelo XXII Governo Constitucional, em resposta à pandemia do *COVID-19*, facto que colocou a empresa *Withus* em regime de teletrabalho desde o início do confinamento.

Sintetizando, o uso de um algoritmo de *Machine Learning* permite otimizar a injeção de energia fotovoltaica numa instalação trifásica.

Este trabalho foi uma experiência provida de motivação e de desafio pessoal. Existiram momentos de algum desânimo acerca da forma como encontrar o fio condutor que possibilitasse a elaboração deste estudo, mas foram sendo ultrapassados pela aquisição e construção do conhecimento, ao longo do tempo, sobre o tema proposto para o estudo.

Salienta-se que foi efetuado um importante percurso de aprendizagem relativamente à vertente teórica da investigação e à pesquisa e utilização de diferentes algoritmos de *Machine Learning* e métricas de erro, os quais poderão refletir-se futuramente na atividade profissional. Além disso, possibilitou o aumento do conhecimento sobre as ferramentas de programação de *Machine Learning* em *Python 3*.

Em contexto real, este trabalho pode servir de base para uma integração ideal da poupança de energia dos painéis fotovoltaicos nos sistemas trifásicos, dependendo do interesse das empresas ou de particulares em adotar esta funcionalidade em relação à comutação de fases destes sistemas.

Por todas estas razões, foi concluído que o programa desenvolvido, ainda que seja um protótipo rude, pode ser aplicado a casos reais de previsão para decisão de fases de instalações trifásicas, para maximizar o aproveitamento de energia fotovoltaica, tanto a empresas, como nos sistemas elétricos de autoconsumo, funcionando em sistemas que contenham *Python 3* instalado.

Considera-se que este estudo apresenta utilidade, já que poderá servir de análise e reflexão para outros trabalhos na mesma área de intervenção, ou noutros contextos futuros de modo mais vasto.

Para terminar, importa referir que, como em qualquer investigação, existem, neste estudo, algumas limitações que condicionaram o resultado e que importa enumerar.

Uma limitação relaciona-se com o facto de os dados não poderem ser extraídos diretamente da plataforma, ou seja, não existe acesso direto aos valores históricos das instalações trifásicas. No entanto, fez-se uso dos dados, já armazenados nos equipamentos de medição de consumo energético, fornecidos pela *Withus*.

Outra limitação refere-se ao facto de os equipamentos de medição de consumo energético só manterem os registos dos últimos três meses, aproximadamente, eliminando constantemente os mais antigos. Deste modo, não foi possível testar a sazonalidade anual.

Por último, podem aparecer *data-sets* da instalação trifásica, contendo intervalos de tempo sem registos de qualquer valor de energia. Estas séries

temporais descontínuas podem resultar em previsões mais desfavoráveis. Apesar de se ter minimizado o problema, escolhendo só o maior intervalo de tempo contínuo a partir desses dados, os resultados podem variar significativamente consoante a duração desse intervalo de tempo.

Como proposta para possíveis trabalhos futuros, poder-se-iam utilizar mais algoritmos e configurações não usados neste trabalho, com a finalidade de obter um maior número de previsões e averiguar se os algoritmos apresentam melhores resultados que os utilizados neste projeto. Além disso, seria importante obter outro tipo de séries temporais, para além da energia reportada, como, por exemplo, a temperatura e a meteorologia, de forma a aumentar a abrangência do problema de seleção de fases de instalação trifásica, resultando em previsões com maior precisão. Seria, igualmente, importante que estes *data-sets* fossem definidos com intervalos de tempo superiores a três meses, de preferência a vários anos, a fim de prever a sazonalidade com maior precisão.

Referências

Alaliyat, S. (2008). Video-based fall detection in elderly's houses [Norwegian University of Science and Technology]. In *Brage.Bibsys.No*. https://www.researchgate.net/publication/267953942%0AVideo

Antunes, A. (2017). *Análise energética de sistemas de abastecimento de água: previsão dos consumos recorrendo a técnicas de inteligência artificial*. Universidade de Aveiro.

Araújo, G. S. de. (2013). *Uso de Random Forests e Redes Biológicas na Associação de Polimorfismos à Doença de Alzheimer*. Universidade Federal de Pernambuco.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, *16*(5), 412–424. https://doi.org/10.1093/bioinformatics/16.5.412

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. http://www.jstatsoft.org/v17/b05/

Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (1977). Time Series Analysis: Forecasting and Control. *Journal of Marketing Research*, *14*(2), 269. https://doi.org/10.2307/3150485

Breiman, L. (2001). *Random Forests*. *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Carugo, O. (2007). Detailed estimation of bioinformatics prediction reliability through the Fragmented Prediction Performance Plots. *BMC Bioinformatics*, *8*(1), 380. https://doi.org/10.1186/1471-2105-8-380

Chen, C., Liaw, A., & Breiman, L. (1999). Using Random Forest to Learn Imbalanced Data. *Discovery*, 1–12.

Criminisi, A., Shotton, J., & Konukoglu, E. (2014). *Decision Forests* (pp. 99–149). https://doi.org/10.1142/9789814590082_0009

Diamantidis, N. A., Karlis, D., & Giakoumakis, E. A. (2000). Unsupervised stratification of cross-validation for accuracy estimation. *Artificial Intelligence*, *116*(1–2), 1–16. https://doi.org/10.1016/S0004-3702(99)00094-6

Dubinets, L. (n.d.). *TOP-DOWN DECISION TREE INDUCERS*.

Ferreira, P. (2010). *Aplicação de Algoritmos de Aprendizagem Automática para a Previsão de Cancro de Mama*. https://www.dcc.fc.up.pt/~ines/aulas/1516/DM1/TesePedroFerreira.pdf

Finlay, S. (2014). Predictive Analytics, Data Mining and Big Data. In *Predictive Analytics, Data Mining and Big Data*. Palgrave Macmillan UK. https://doi.org/10.1057/9781137379283

Francisco, S. I. M. (2015). *Recognition of Cancer using Random Forests as a Bag-of-Words Approach for Gastroenterology*. Universidade do Porto.

Furão, S. da S. (2018). *Criação de Interfaces Gráficas Automatizadas, Dinâmicas e Adaptáveis*. Universidade de Aveiro.

Harvey, A. C., & Peters, S. (1990). Estimation procedures for structural time series models. *Journal of Forecasting*, *9*(2), 89–108. https://doi.org/10.1002/for.3980090203

Hripcsak, G., & Adam S. Rothschild. (2005). Agreement, the F-Measure, and Reliability in Information Retrieval. *Journal of the American Medical Informatics*

Association, *12*(3), 296–298. https://doi.org/10.1197/jamia.M1733

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15. http://arxiv.org/abs/1412.6980

Liu, D. C., & Nocedal, J. (1989). *On the limited memory BFGS method for large scale optimization*.

Lopes, C. R. (2019). *Utilização de Modelos Estatísticos e Machine Learning para a previsão de vendas no Setor de Retalho – Um Estudo Comparativo*.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLOS ONE*, *13*(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

Martins, I. F. S. (2011). *Machine Learning Algoritmhs to Predict Blood-Brain Barrier Permeability of Drug Molecules*. Universidade de Lisboa.

Mitchel, T. M. (1997). *Machine Learning*. McGraw-Hill. https://doi.org/10.1007/978-3-642-21004-4_10

Oshiro, T. M. (2013). *Uma abordagem para a construção de uma única árvore a partir de uma Random Forest para classificação de bases de expressão gênica*.

Ramos, P., Santos, N., & Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and Computer-Integrated Manufacturing*, *34*, 151–163. https://doi.org/10.1016/j.rcim.2014.12.015

Renaud, O., & Victoria-Feser, M.-P. (2010). A robust coefficient of determination for

regression. *Journal of Statistical Planning and Inference*. https://doi.org/10.1016/j.jspi.2010.01.008

Rijo, S. M. A. (2017). *Técnicas de Deep Learning para Detecção de Eventos em Áudio*. Universidade de Évora.

Silva, J. C., Farias, F. C., Lima, V. C. F., Silva, V. L. B., Seijas, L. M., & Bastos-Filho, C. J. A. (2015). Classificação de Sinais de Trânsito Usando Otimização por Colmeias e Random Forest. *Anais Do 12. Congresso Brasileiro de Inteligência Computacional*, 1–6. https://doi.org/10.21528/CBIC2015-166

Tan, P.-N., Steinback, M., & Kumar, V. (2003). *Introduction to Data Mining*.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, *16*(4), 437–450. https://doi.org/10.1016/S0169-2070(00)00065-0

Taylor, S. J., & Letham, B. (2017). Business Time Series Forecasting at Scale. *PeerJ Preprints 5:E3190v2*, *35*(8), 48–90. https://doi.org/10.7287/peerj.preprints.3190v2

Theocharides, S., Makrides, G., Georghiou, G. E., & Kyprianou, A. (2018). Machine learning algorithms for photovoltaic system power output prediction. *2018 IEEE International Energy Conference (ENERGYCON)*, 1–6. https://doi.org/10.1109/ENERGYCON.2018.8398737

Tong, J. C. (2013). Cross-Validation. In *Encyclopedia of Systems Biology* (2nd ed., pp. 508–508). Springer New York. https://doi.org/10.1007/978-1-4419-9863-7_941

Vieira, P. G. F. (2017). *Deep learning para identificação de mutações genéticas patogénicas*. Universidade de Aveiro.

Wan, C., Zhao, J., Song, Y., Xu, Z., Lin, J., & Hu, Z. (2015). Photovoltaic and solar power forecasting for smart grid energy management. *CSEE Journal of Power and Energy Systems*, *1*(4), 38–46. https://doi.org/10.17775/CSEEJPES.2015.00046

Weisberg, S. (2005). *Applied Linear Regression* (3rd ed.). John Wiley & Sons, Inc. https://doi.org/10.1002/0471704091

Yona, A., Senjyu, T., & Funabashi, T. (2007). Application of Recurrent Neural Network to Short-Term-Ahead Generating Power Forecasting for Photovoltaic System. *2007 IEEE Power Engineering Society General Meeting*, *39*(15), 1–6. https://doi.org/10.1109/PES.2007.386072

[1] http://withus.pt/ Acedido a 15 de maio de 2020

[2] https://medium.com/@dhiraj8899/top-5-difference-between-linear-regression-and-logistic-regression-893f6470d7e0 Acedido a 18 de maio de 2020

[3] https://mlfromscratch.com/machine-learning-introduction-8-linear-regression-and-logistic-regression/ Acedido a 5 de junho de 2020

[4] https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 Acedido a 5 de junho de 2020

[5] https://mc.ai/a-to-z-about-artificial-neural-networks-ann-theory-n-hands-on/ Acedido a 4 de junho de 2020

[6] https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 Acedido a 16 de maio de 2020

[7] https://towardsdatascience.com/forecasting-stock-prices-using-prophet-652b31fb564e Acedido a 18 de maio de 2020

[8] https://www.dataschool.io/comparing-supervised-learning-algorithms/ Acedido a 7 de junho de 2020

[9] https://facebook.github.io/prophet/ Acedido a 16 de maio de 2020

[10] https://www.python.org/download/releases/3.0/ Acedido em 15 de maio de 2020

[11] https://www.scipy.org/ Acedido a 15 de maio de 2020

[12] https://numpy.org/ Acedido a 15 de maio de 2020

[13] https://pandas.pydata.org/ Acedido a 15 de maio de 2020

[14] https://matplotlib.org/ Acedido a 17 de maio de 2020

[15] https://devdocs.io/scikit_learn/ Acedido a 16 de maio de 2020

[16] https://www.geeksforgeeks.org/learning-model-building-scikit-learn-python-machine-learning-library/ Acedido a 26 de maio de 2020

[17] https://www.arundo.com/arundo_tech_blog/adtk-open-source-time-series-anomaly-detection-in-python Acedido a 15 de maio de 2020

[18] https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/ Acedido a 18 de maio de 2020

[19] https://github.com/raimas1996/three-phase_energy_comuter Acedido a 6 de junho de 2020

[20] https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ Acedido a 18 de maio de 2020

[21] https://www.tensorflow.org/guide/keras/train_and_evaluate Acedido a 18 de maio de 2020

[22] https://scikit-learn.org/stable/modules/tree.html Acedido a 18 de maio de 2020

[23] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html Acedido a 16 de maio de 2020

[24] https://pythondata.com/forecasting-time-series-data-with-prophet-part-1/ Acedido a 18 de maio de 2020

[25] https://docs.python.org/3/howto/sockets.html Acedido a 6 de junho de 2020